2008-12

A comparative analysis of wiki discretionary access control in a CONOPS environment

Crawford, Frederick L.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/3865

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A COMPARATIVE ANALYSIS OF WIKI DISCRETIONARY ACCESS CONTROL IN A CONOPS ENVIRONMENT

by

Frederick L. Crawford

December 2008

Thesis Advisor:	Karl Pfeiffer
Second Reader:	Rex Buddenburg

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2008	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE A Comparative Analysis of Wiki Discretionary Access Control in a CONOPS Environment	**5. FUNDING NUMBERS**
6. AUTHOR(S) Frederick L. Crawford	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This research conducts a comparative analysis of discretionary access controls of current wikis by experimenting with their discretionary access controls and functionality, comparing the wiki software cost of implementation, and comparing the scalability for possible enterprise use. Most importantly, the author will analyze wikis discretionary access control capabilities and suitability in regards to which wiki will be more beneficial in a particular CONOPS.

The derivation of the author's thesis focuses awareness on effective information allocation that is reliable and accurate while maintaining its confidentiality based upon some level of discretionary access control (DAC). In the author's opinion, wiki technology enables near real-time information, fosters Communities of Practice (CoP), enhances collaboration, and reduces information stovepipes. The author will examine different wikis to determine which wiki DAC implementations are most suitable for different CONOPS objectives. To determine the best wiki complement with CONOPS objective, the author will conduct tests and a comparative analysis. The comparative analysis consisted of DAC mechanisms and administrator functions.

14. SUBJECT TERMS Wiki, Collaboration, CONOPS, Comparative Analysis	**15. NUMBER OF PAGES** 95
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A COMPARATIVE ANALYSIS OF WIKI DISCRETIONARY ACCESS CONTROL IN A CONOPS ENVIRONMENT

Frederick L. Crawford
Lieutenant, United States Navy
B.S., Norfolk State University, 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(COMMAND, CONTROL, and COMMUNICATIONS (C3))

from the

NAVAL POSTGRADUATE SCHOOL
December 2008

Author: Frederick L. Crawford

Approved by: Karl Pfeiffer
Thesis Advisor

Rex Buddenburg
Second Reader

Dan Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This research conducts a comparative analysis of discretionary access controls of current wikis by experimenting with their discretionary access controls and functionality, comparing the wiki software cost of implementation, and comparing the scalability for possible enterprise use. Most importantly, the author will analyze wikis discretionary access control capabilities and suitability in regards to which wiki will be more beneficial in a particular CONOPS.

The derivation of the author's thesis focuses awareness on effective information allocation that is reliable and accurate while maintaining its confidentiality based upon some level of discretionary access control (DAC). In the author's opinion, wiki technology enables near real-time information, fosters Communities of Practice (CoP), enhances collaboration, and reduces information stovepipes. The author will examine different wikis to determine which wiki DAC implementations are most suitable for different CONOPS objectives. To determine the best wiki complement with CONOPS objective, the author will conduct tests and a comparative analysis. The comparative analysis consisted of DAC mechanisms and administrator functions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AOC	Air Operations Center
ATO	Air Tasking Order
CFACC	Coalition Forces Air Component Commander
CFLCC	Coalition Forces Land Component Commander
CFMCC	Coalition Forces Maritime Component Commander
COCOM	Component Commander
CoP	Community of Practice
CVS	Concurrent Version System
DAC	Discretionary Access Control
DJIOC	Defense Joint Operations Center
DoD	Department of Defense
GWOT	Global War on Terrorism
JIOC	Joint Intelligence Center
JIOC-X	Joint Intelligence Operations Capability-Transformational
JTF	Joint Task Force
MLS	Multilevel Security
NCSC	National Computer Security Center
OPSEC	Operational Security
RCS	Revision Control System
TACTICOMP	Tactical Personal Digital Assistance
ZCS	Zimba Collaboration Suite

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to express his sincere thanks to the many individuals who have assisted throughout the course of this thesis study.

For starters, the author would like to thank Dr. Karl Pfeiffer and Professor. Rex Buddenburg for their time and effort in helping to formulate the scope of this thesis and their assistance throughout the length of this research. Dr. Pfeiffer's words of wisdom, leadership and genuine concern made the academic challenge achievable and obtainable in a timely manner. Professor Buddenburg's guidance and encouragement enable academic growth. The author appreciates your patience and constructive comments during this thesis.

In addition, the author would like to thank Mr. Phil Hopfner and Mr. Buddy Barreto for their technical expertise and assistance in helping with troubleshooting issues encountered during the implementation process. The author is especially grateful to Mr. Hopfner for his patience and time in going through installation instructions and configuration procedures to make them error-free.

Furthermore, the author would also like to thank CAPTAIN Kathryn Hobbs, United States Navy, Navy Postgraduate School Dean of Students, for her words of encouragement through some very challenging times. Her words of encouragement provide great insight—there is always light at the end of the tunnel.

Last but not less, the author would like to share the success of the completion of this thesis research with family, close friends, and God almighty because, without their unwavering support and encouragement, this thesis would not have been possible.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. OVERVIEW

In the past, military command and control (C2) was constrained to single agency operations in which information sharing between a commander and his subordinates was organized in a hierarchical structure in order to simplify planning and controlling functions. Information systems used by commanders in the military have been constrained by the communication technology limitations: a company commander had a radio (or wire or a messenger) that would connect him to the battalion headquarters… and not much else. In general, our communications systems followed the hierarchy of our C2 structure. One exception to this rule is the radio: as a shared medium, this is often used with non-government organizations and coalition partners, flattening the hierarchy somewhat. Our communication flow should support but not necessarily mimic our military structure. Connectivity is engrained in this hierarchy, as information gathering and the passing of that information to higher levels are procedures associated with centralized management of the battlespace.

As communication flows, some form of filtering, adding, deleting, and modification is done at each level. This editing is time-consuming and can often result in the critical information not reaching the right people, or getting there too late. In attempting to get the right information to the right people on time, some degree of freedom is required at all levels to better balance decision-making. This increased volume of information supports faster decision-making to keep up with the increased tempo of warfare.

Military command and control spans geographical boundaries, as well as agency, coalition, and allied information domains. The need to collaborate among interested parties presents an interesting challenge. ”It requires increased collaboration and cooperation between and among individuals and organizations who are interested in defense transformation in general, and specifically it requires new C2 approaches that anchor coevolved network-centric mission capability packages [32].”

Perhaps our hierarchical organization and control of information hinders our ability to accomplish this objective [18]. The use of web-based, collaborative technology, wikis, may help to flatten the information structure supporting today’s command and control structure. Wikis would allow vital processed information to be shared and delivered to the tactical user for faster decision-making on the battlefield.

B. PURPOSE

The objective of this research is to explore the discretionary access control common among web-based collaborative technologies, using the wik as our example. This thesis will address these questions:

1. Is the wiki paradigm a useful concept for military collaboration?
2. Do the access controls within wiki implementations support necessary hierarchical controls on current information domains of interest to the military?

To address these questions in depth, this research will examine MediaWiki and TWiki implementations, testing their discretionary access control (DAC) features within the context of an operationally relevant scenario.

C. ORGANIZATION OF PAPER

This rest of this thesis is organized as follows:

- Chapter II provides background information on the wiki as a collaborative information technology; and on discretionary access control as an information assurance tool.

- Chapter III discusses experimental methodology, to include scenario development; criteria for wiki selection; and test plan for experimentation.

- Chapter IV provides the comparative analysis on selected wiki engines based on their DAC implementations and suitability for a CONOPS environment. We conclude with suggestions on how a wiki might be employed in a particular CONOPS scenario.

- Chapter V presents a summary of the work and conclusions drawn from this research, with emphasis on suggestions for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

This chapter provides the academic framework for our research. We being with a description of the wiki collaborative technology, then examine the discretionary access controls attendant to two specific wiki implementations: MediaWiki and TWiki.

A. WIKI

Technological advances in computers and the ubiquity of the Internet have facilitated the way we communicate and share information. These tools have become a cornerstone of operational decision-making and are used to enhance collaboration during planning and development. Collaboration software facilitates the collecting, refining and sharing of explicit *and tacit* knowledge among many communities of practice (CoP).

A wiki is a collaborative software tool that enables anyone to contribute information through a web-based service. More simply, a wiki is web-based software that allows all viewers of a page to change the content by editing the page online using a browser interface [3]. Individual contributions can be easily added, edited, changed or deleted by other community of interest workers. Wikis represent a form of open information sharing and as such represent a powerful and promising collaboration tool.

1. Collaboration

Collaboration is the process of interaction among participants with shared or congruent goals. In the mainstream literature, however, this is an ambiguous term and merits more careful definition for this research. Collaboration in the workplace, for example, can be among individuals, teams or whole enterprises. It can be synchronous, that is, between people who must be available at a particular time; or asynchronous, where the communicating parties do not need to be present simultaneously. It can be ad hoc or structured, in the same way that much of an organization's information may be held in both unstructured and structured data [10].

Structured collaboration represents a process that is well understood and, to a large extent, can be predicted. Ad hoc collaboration cannot be predicted in terms frequency or content. In crisis response, in particular in military operations, much collaboration is ad hoc, and in terms of this research unstructured collaboration is often of the greatest value to the mission. One aspect of ad hoc collaboration is the pooling and generation of new ideas; this is particularly powerful in responding to an evolving battle space.

2. History

The wiki concept is often credited to Ward Cunningham, a software engineer from Portland, Oregon, working in 1995 in object-oriented design and programming [2]. As a programmer, he often grappled with communicating complexity when sharing documents and programs with other programmers; these challenges inspired him to develop a collaborative tool that would be suitable for his working environment [31]. Moreover, his desire to develop a simple collaborative software tool came to fruition through his development of a collaborative software framework: the wiki. "Wiki wiki" is a Hawaiian word that means "quick or hurry," and in common use today the term wiki indicates quick, mass collaboration [1].

3. Advantages and Disadvantages of Using a Wiki

The function a wiki serves comes from the requirements of the community or organization supported. Often, wikis serve as knowledge management tools for a community of practice (CoP). For this research, we will view the wiki in this manner.

The common user interface of the wiki, available to any web-based client, coupled with a simple, time-tested markup language, can reduce the need for printed or mass-distributed organizational publications or instructions. Most important for modern clients (e.g. Blackberries, iPhones, tactical digital assistants like the TACTICOMP), wikis do not require extensive user training or loading of applets, making the tool less complex to support and to use [24].

A significant disadvantage with the wiki is that the information management tool is as effective as its community of practice: if not properly monitored and maintained, users could easily input useless information and render the wiki ineffective. In concert with this problem is that of editing rights and authentication: maintainers or web administrators who have full control rights would have access to sensitive files with no premise of need-to-know justification. Although potential useless information and authenticity remain an issue, this signal to noise in a wiki is largely a function of community vigilance [24].

4. How Wiki Software Works

The technical details of how a wiki works are simple but fascinating. Wiki software is installed as a script, which resides on a server. Once on a server, small documents are produced, called *wiki pages* or articles, which are accessible through any web browser. For example, when a wiki based Internet page is accessed, a query is sent to the server where the wiki software resides. The data is in the form of simple text, which needs to be formatted in order to display in the browser. Next, wiki scripts translate the wiki code into HTML and embed it in the wiki page to be sent to the browser [1]. The wiki script can come in many forms such as (PHP) scripts, which read the raw page data from a MySQL database or a flat file. The data is then checked, line by line, and the specific format commands contained in it are replaced by the matching HTML codes. Each page is identified by its distinct subject name, commonly linked in the navigation menu [1].

5. Wiki Functions

In general, wiki software contains five functions: edit; link; history; recent change; and search. These functions enable participants to effectively use wiki and are discussed below.

While maintaining a wiki, the most used function is the ‘Edit’ function. When editing a page, a query similar to a read request is sent to the server, though the returned page is not converted to HTML format by the web-browser interface. Rather, the raw HTML for the page is returned for modification by the user editor. This web-based client editing gives the user (community member) the ability to update information and replace a version in the wiki database with new information. Reading users will see these changes when the wiki entry is refreshed [1].

Key to the usefulness of the wiki is the development of knowledge from information. In implementation, this is accomplished with the Link function, which permits similar or related articles to be linked to one another. These tacit relationships become explicit metadata in the wiki structure.

The History function saves all previous versions or modifications of a particular page. This function aids in tracking the activities of users who are adding, deleting or editing the page, and leaves time stamps associated with these changes. The History function allows the administrator or community to police or block users who may add information that is malicious in nature, and allows administrators to roll back to a prior version. The History function is reserved for users with administrative rights only. Moreover, the History function is vital to data integrity and information assurance within the wiki.

This history function is similar to revision control systems used for software development. One current but time-tested example is the Concurrent Version System (CVS). CVS allows multiple users to work with a single document simultaneously without loss of data. CVS utilizes a similar tracking method used in a wiki, an underlying, granular revision control system or RCS [1].

The Recent Change function provides a current overview of a certain number of recent changes to wiki pages or all changes within a predefined period [1]. This function uses software called *watch lists*. The watch list monitors selected pages without requiring the administrator to the conduct the cumbersome task of searching each page or article for changes.

A ‘Sandbox’ serves as the training space for new or in experienced users. In addition, it offers user-friendly instructions and tutorials about basic wiki usage and an empty wiki page for experimentation prior to use of a regular wiki page.

Finally, the ‘Search’ function allows users to quickly access pages or articles associated with the wiki. For instance, titles function as keys like that in a database; the presence of relevant keywords in titles will tend to make them “well-written” and responsive to the search function. The search function can work like many other search functions in search engines, such as Google or Yahoo, allowing users to find or access information quickly without strolling through the entire document.

6. Wiki and Web Administrator

Two distinct administrative roles are required to support a wiki: wiki administrators and web administrators. Both will be discussed briefly. To maintain wiki usability, a wiki administrator has to ensure that the day-to-day operation is running smoothly. Entrusted with more control and permissions than regular users, wiki administrators or system administrators have overarching responsibility for policing regular users and their content. Moreover, they have the authority to delete and deny access to any regular user [1]. Wiki administrators have the responsibly to appropriately deal with acts of vandalism and deal with revert wars. They also have the ability to rollback the versions in case of vandalism in seconds if required. They usually have their own interface to the wiki to which they only have access.

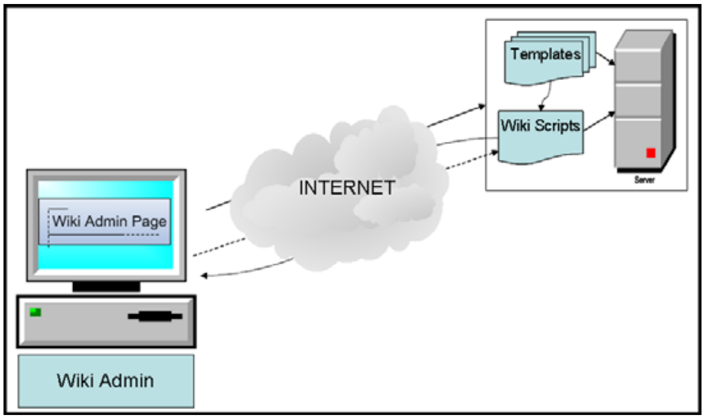


Figure 1. Wiki Administrator (From: [1])

A web administrator is the keeper of the wiki software, and is responsible for its maintenance and updates. Web administrators have direct access to the server and files without having to be a member of the community of practice associated with the wiki content. Therefore, web administrators would need to have super user or full control access to carry out day-to-day tasks.

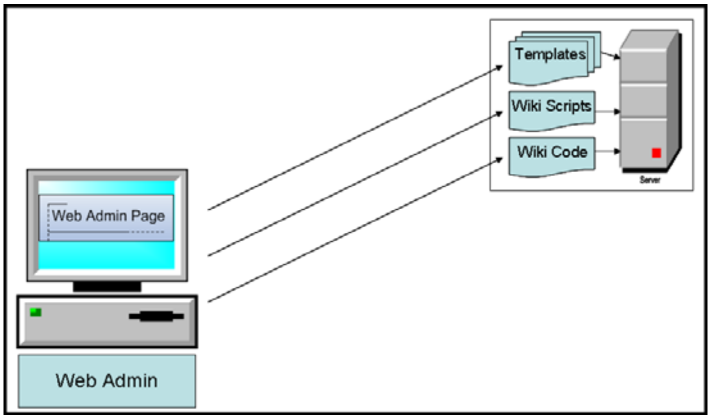


Figure 2. Web Administrator (From: [1])

7. Community of Practice

The term *community of practice (CoP) was* first defined by Jean Lave, a social anthropologist, and Etienne Wenger, an educational theorist [3]. They defined the term as being "groups of people who share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis [3].”

In addition, “CoP is a process of social learning that transpires when subjects collaborate to contribute ideas, resolutions, and construct innovations [3].” Community of Practice is established by regular interactions. In a military context, community examples might focus on traditional Navy designators (e.g., Intelligence, Aviation, and Surface Warfare) or combatant command areas (e.g., Europe, the Pacific, Korea).

Communities of Practice enable a group of individuals with similar backgrounds, with a similar issue or common problem, to consolidate knowledge underneath one intellectual umbrella. The wiki as a collaboration tool is may be well-suited to serve communities of practice, and in the course of this research we will examine its suitability for military CoPs.

B. DISCRETIONARY ACCESS CONTROLS (DAC)

According to guidance by the National Computer Security Center in Fort Meade, Maryland, “discretionary access control is simply a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a user or process given discretionary access to information is capable of passing that information along to another subject [30].” For clarification, an *object* is a passive entity that contains or receives information, examples are, pages, files or directories [3]. For example, if user B has read permission to a document and after reading determines that users, C, D, and E need to see the information as well. User B could then pass those rights to users C, D, and E only if user B had control permissions of that document.

Many information managers within the DoD enterprise are concerned about the sharing of information and its security. The open sharing intrinsic to a wiki is a significant point of contention in using this technology for DoD communities. Within this research effort, then, we need to examine DAC controls and their role in information protection. Many trusted systems enforce discretionary policies with respect to sharing and retrieving information. DAC is a means of restricting access to objects based on the identity of subjects and/or groups to which they belong [3].

In addition to managing access by metadata or role (e.g., the use of login IDs and passwords), DAC is a very common form of managing access to directories or files. Moreover, DAC and login are thought of as, perimeter control mechanisms that put barriers around information to keep unauthorized users from accessing that information. Therefore, DAC is thought of as, a system of walls within a computer’s file system. According to the National Computer Security Center, on discretionary access control, there are four common modes of DAC—Hierarchical, Ownership, Laissez-Faire and Centralized [2].

- Hierarchical control is familiar to most businesses when securing trusted systems. This form of control implies there is an administrator who will have the overall control to all objects on the system. The administrator could pass control to other users (departments) from the top down within the organization. Hierarchical control also assumes that a superior will have the ability to see all of his subordinates files with the capability of control down and visibility up. This delegation of control depicts the organizational structure of a company or chain of command aboard a ship [2].

- Ownership control implies that whoever creates the document or object is the only individual able to grant access rights to other users. In general, one cannot pass ownership control to objects to other users, but can grant access to directories and files that he created. Some systems provide a mechanism whereby the ownership of a file or directory can be assigned to a different user. In real world systems, the administrator will still be able to obtain full control and grant permissions pertaining to the object created by the owner.

- The laissez-faire control scheme allows anyone who has control permission to pass that permission to other users. This scheme implies that no one owns the document or object.
- The centralized control model gives control to an administrator who has full control to all documents or objects on the system. In this control scheme, a user cannot pass control to other users. Each user needing access would have to request access from the administrator.

From the NCSC perspective, there are five different DAC mechanisms: capabilities, profiles, access control list (ACLs), protection bits, and passwords [2]. A DAC mechanism's *capabilities* protect objects and specify the access rights allowed for access based upon who possesses the capability. For example, users may have the capability to conduct read and write operations onto an object. *Profiles* allow access to a list of protected objects associated with each user. However, if a user has access to a list of protected objects, the profile can be too large and, therefore, difficult to manage, requiring profile updates. This would, in turn, take time for each profile to be checked to gain access to an object. ACLs are associated with objects and contain a list of users or groups and corresponding rights each of these has to be the object. For a given user, only the entry associated with that user must be checked. This mechanism saves time by automating that burdensome process. *Protection bits* are associated with protection of the objects. For example, operating systems such as UNIX use protection bits to verify whether a group or owner has access to protect the object [2]. Finally, *password protection* of objects gives access to anyone with the password; this is available, for example, with Microsoft Word documents [2].

The concepts of control and access permissions are conceptually separate when referring to DAC. *Control permissions* mean having control over objects and being able to pass that control to other users. Examples of control permissions are hierarchical, ownership, laissez-faire, and centralized [2].

Access permissions are a finer granularity of access to a directory or file. There are different configurations in which access can be granted. For example, either on directories and files, or on directories and no files or vice versa. Examples may vary from one system to another, but basic permissions are *read* and*, write access permissions* [2]. For files, an additional permission is often executed.

The importance of tracking how directories and files are stored is fundamental to how discretionary policies are implemented. This process can be, in many cases, very hard to manage for a systems administrator. The management of files or directories aids administrators in how we protect, share, and give permissions to files, in order to ensure the right people access the right information. One major concern is how permissions are passed and controlled. For example, Windows Server 2003 permissions are inherited by default from the root when subdirectories and files are created. Root level access control provides for better manage to these subdirectories or files, Windows Server 2003 allows the administrator the flexibility of changing the default setting. On older revisions, subdirectories and files under a root are sometimes called an *extended directory control* [2]. UNIX operating systems will inherit controls at the root to subdirectories and files under it, but Microsoft's operating system does offer greater flexibility then older versions used in TWiki for management of DAC permissions suitable for a secure environment, such as Concept of Operation (CONOPS). CONOPS is the operational sequence of events pertaining to a mission [14].

C. SUMMARY

This chapter provided an overview of the wiki, with emphasis on the technical requirements that will be needed in a military environment. Toward that end, we also discussed discretionary access controls (DAC) and their common requirements and pitfalls. In concert, the suitability of DAC in specific CONOPS will be illustrated. We next explore two wiki implementations, MediaWiki and TWiki, in the context of an operationally relevant scenario.

III. METHODOLOGY AND EXPERIMENTATION

A. WIKI SELECTION

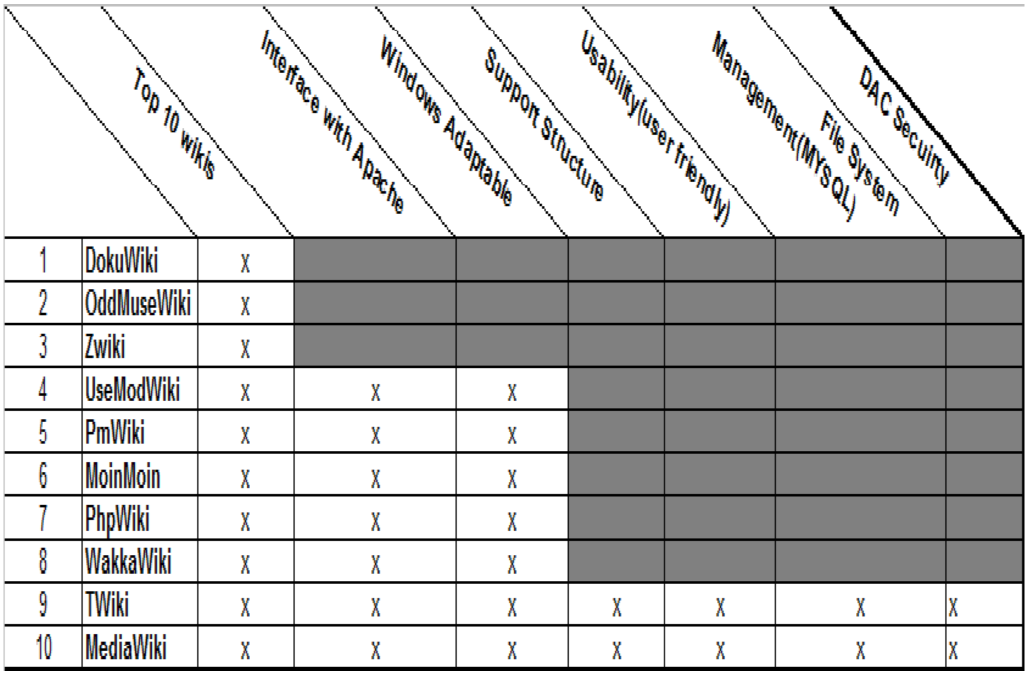
There are over 140 wiki engines available online, and many more are under development [19]. Wiki engines offer a wide range of functions and versatility. Because our goal is to demonstrate the suitability of the wiki paradigm and not advocate for any particular engine, we went through a short selection process to find two suitable examples. This chapter describes the engines selected and the plan of experimentation for demonstrating their use in a military CONOPS development.

1. Selection Methodology

The widespread use of wiki technology and the benefits of its capabilities are starting to grow among many business enterprises. A comparative analysis of wiki engines is available on the Internet. However, for the purpose of this research, a systemic approach was taken to narrow a list of 140 wikis to 10 that may be applicable to a CONOPS environment [19].

2. Selection Process

The criteria utilized for wiki selection was a two-step process. The first process addresses three criteria dealing with wiki engine capabilities and the second process addresses three criteria dealing with CONOPS requirements. The selection of wiki engines suitable for a CONOPS environment was taken from 10, based on a top-ten list available online and shown in Table 1 [24].



Top 10 wikis		Interface with Apache	Windows Adaptable	Support Structure	Usability(user friendly)	Management(MYSQL)	File System	DAC Security
1	DokuWiki	x						
2	OddMuseWiki	x						
3	Zwiki	x						
4	UseModWiki	x	x	x				
5	PmWiki	x	x	x				
6	MoinMoin	x	x	x				
7	PhpWiki	x	x	x				
8	WakkaWiki	x	x	x				
9	TWiki	x	x	x	x	x	x	x
10	MediaWiki	x	x	x	x	x	x	x

Table 1. Wiki Selection Process Results

a. Wiki Capabilities Criteria

(1) Software Interface. The majority of wikis use the Apache Hypertext Transfer Protocol Server to edit and serve content, and so an Apache-based wiki seemed a reasonable choice for this experimentation. Since the DoD network environment is often diverse, another consideration is whether the wiki engine can be interoperable with other commonly used software, in particular Windows products.

(2) Maintainability. The second consideration, maintainability, is extraordinarily important for DoD operations. Wiki engines that have a solid support structure of five or more support groups (commercial-support business model) will ensure that patches are up-to-date to mitigate growing security threats, and prevent system failure, and/or loss of data [19].

(3) Adaptability. The last criterion addressed the wiki engine functionality and adaptability. For example, DoD CONOPS can be

generated by many different communities with different levels of operating security. Thus, the wiki engine should be able to adapt to different CONOPS security, and cultural challenges.

b. CONOPS Criteria

From our ten candidates (Table 1), only two wiki engines met the necessary requirements to test the CONOPS scenario. Requirements needed for the DoD CONOPS environment were usability, file system management capabilities, and discretionary access security controls.

(1) Usability. With today's information-intensive environment and user-friendly software leads to greater acceptability. Usability thus fosters greater productivity within an organization when employees feel comfortable with software that will help them easily complete daily tasks. For example, according to the TWiki web site, Eric Baldeschwieler, Director of Software Development of Yahoo, stated:

> Our development team includes hundreds of people in various locations all over the world, so web collaboration is VERY important to us. TWiki has changed the way we run meetings, plan releases, document our product and generally communicate with each other. We're great fans of your work. [21]

(2) File System Management. Within the DoD enterprise, there are a myriad of documents, both classified and unclassified, that are tracked and audited on a regular bases. The use of a standard file management system capability that possesses some form of access control is paramount.

(3) Discretionary Access Controls. The last and most important criterion is support for discretionary access control capabilities. More specifically, we need to examine whether the wiki security mechanisms allow for fine granularity in managing access to wiki content.

c. Wiki Selection

MediaWiki is used extensively by not-for-profit and non-profit organizations [1]. The features and functions do not require users to learn any programming or take any specific classes before using. Moreover, its standard file system is a scalable, feature-rich wiki engine that uses PHP to process and display data stored in a MySQL database [20]. Furthermore, MediaWiki has the capability to manage and store millions of images and multimedia files. Most importantly, it has a robust DAC capability that allows permissions to be assigned to a particular CoP or group. It allows administrators or users to apply fine granularity permissions to files or pages on a need-to-know basis [20].

TWiki is the most popular with many businesses running enterprise-based networks [21]. TWiki is a powerful enterprise file system, which utilizes wiki technology for enterprise collaboration and is interoperable with knowledge management systems. Additionally, TWiki uses the Perl scripting language; it aids in the flow of information within an organization, and promotes distributive teamwork across geographical domains in a seamless environment [22]. Most importantly, it also has a robust DAC permissions capability that allows groups to be established. TWiki enables CoP to have flexibility in assigning permissions to share or protect files from non-members of a particular group. Table 3 displays the results from the selection process [22].

IV. CONCEPT OF OPERATION SCENARIOS

This section describes the scenarios developed to test MediaWiki and TWiki. Today's planning approaches in a network centric environment are simply not suitable to meet today's missions and challenges [27]. To accommodate the need for the increased tempo in decision-making, the need for greater "speed of command and control" will require a more effective collaborative tool to support decision-making. Today, there are many improved information systems supporting military organizations. This increases the frequency and volume of information decision makers want in expressing the commander's intent when conducting military concept of operations (CONOPS) [14]. Furthermore, those improved systems will also contribute to flattening of the command and control structure. Flattening is reducing communication levels within the chain of command; as results, the end-user receives information faster.

The operational "scheme of maneuver" describes all sequences of events pertaining to the mission, and is called a Concept of Operations [14]. Military CONOPS entails methodical detailed planning and a comprehensive evaluation of all objectives or tasks. Within this detailed planning structure, decision makers can retrieve, collect, disseminate, evaluate, and process vital information needed for coordinated actions. A CONOPS permit each decision maker to formulate mission objectives and meet operational objectives to suit their desired end state. This planning process is extremely important when dealing with coordinated military operations and interagency collaboration. Moreover, joint CONOPS among Non-governmental Organizations (NGO) and International Organizations are particularly essential to mission success in response to environmental disasters or humanitarian relief missions.

We next describe two scenarios: a real-life CONOPS situation and a hypothetical CONOPS scenario. These scenarios will be used to examine the suitability of the wiki as a collaboration tool among DoD communities.

A. CONOPS — JOINT INTELLIGENCE OPERATIONS CAPABILITY — TRANSFORMATIONAL (JIOC-X)

The Chairman, Joint Chiefs of Staff, directed each Combatant Commander (COCOM) to stand up a Joint Intelligence Operations Center (JIOC) that will integrate intelligence, operations, and plans in order to, plan, prepare, integrate, direct, synchronize, and manage continuous, full-spectrum defense intelligence operations [16]. To support this endeavor, a transformational Joint Intelligence Operations Capability (JIOC-X) was established under USJFCOM to facilitate this process [16].

This section will provide background about the JIOC-X and will describe how a wiki might be employed within the JIOC-X.

1. JIOC-X Overview

Post-analysis of the events of 9/11 highlighted the need for collaboration among DoD and national intelligence agencies [27]. In trying to address this shortfall, the intelligence community recognized many imminent challenges. For example, this end state required overarching integration with agencies to remove cultural, institutional, and technological stovepipes. This however calls for a paradigm shift in top-down hierarchical command and control structure to a command and control structure that removes levels of processes within the organizational topology. Such removal would enable vital information to be shared and delivered in real time to both senior decision makers and operational forces. The focus of the JIOC-X will be to assist Defense Joint Operations Center (DJIOC) and COCOM JIOCs in leveraging full capabilities towards the integration of plans, operations and intelligence [16].

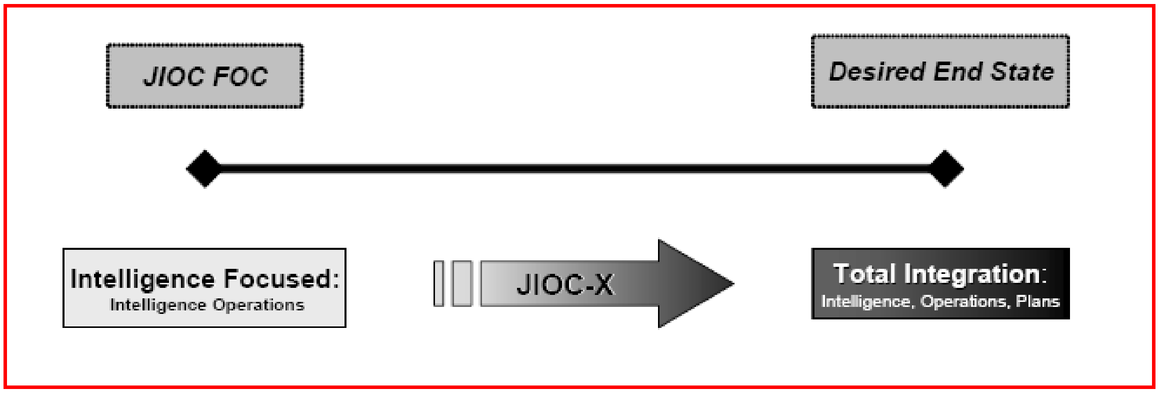
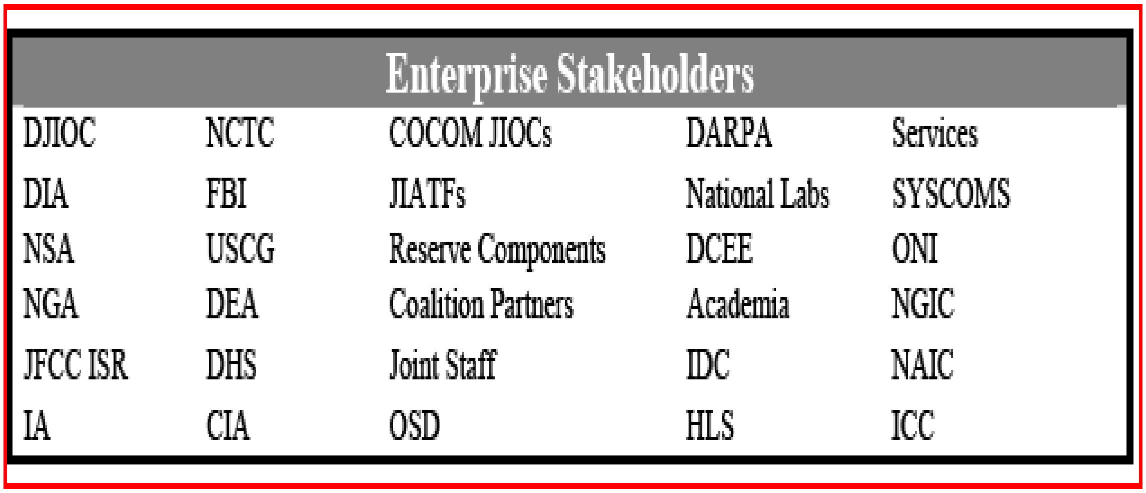


Figure 3. JIOC-X Transformation (From: [16])

2 JIOC-X Wiki Implementation

This JIOC-X CONOPS will require collaborative technology, such as wikis, to meet this desired end-state. The open sharing possible with a wiki framework would seem well-suited to this task. For example, traditional intelligence is processed from the bottom up; it has to travel back down through the chain of command to the end user. A wiki helps to shorten this time-consuming process by keeping the end users and top decision makers in near-real-time communication. However, the fundamental innovation behind a wiki is not the technology, but its ability to promote sharing. The requirements for such technology will require scalability, usability, and the ability to integrate in a secure operational environment. Although the sharing features of a wiki may seem suffice, a comprehensive analysis of the JIOC-X security requirements will be explored using a hypothetical scenario to determine if wiki discretionary access control provides suitable protection of information. The DAC permissions discussed in Chapter II and the systematic instructions for setting permissions are essential to the use of wikis in a JIOC-X CONOPS.

Consider the following hypothetical scenario: JIOC-X has established a wiki in which each participating partners can share operational plans and intelligence in support of ongoing operations in the Global War on Terrorism (GWOT). Figure 4 depicts all the key players involved.



Enterprise Stakeholders

DJIOC	NCTC	COCOM JIOCs	DARPA	Services
DIA	FBI	JIATFs	National Labs	SYSCOMS
NSA	USCG	Reserve Components	DCEE	ONI
NGA	DEA	Coalition Partners	Academia	NGIC
JFCC ISR	DHS	Joint Staff	IDC	NAIC
IA	CIA	OSD	HLS	ICC

Figure 4. Key Players (From: [16])

In order to break down cultural and institutional barriers, JIOC-X will serve as the wiki administrator. Control rights will be configured in hierarchical discretionary access control architecture. Control rights are activated at the parent directory to increase collaboration between participating entities within one particular COCOM JIOC. Furthermore, this form of collaboration will enhance combat effectiveness and break down cultural barriers associated with many communities.

B CONOPS INTER-SERVICE FOR JOINT TASK FORCES PACIFIC TEAK (JTF TEAK)

Joint planning and operations among the different services continues to be a complex challenge. Community stovepipes, parochial lines of communication established firmly in the hierarchy, make information collaboration particularly complex among organizations. While there are policy issues that need to be addressed, the focus of this thesis will highlight the use of wiki technology and its implementation in achieving total integration. The following scenario described is hypothetical and used to illustrate wiki implementation in a secure CONOPS environment [17].

1. JTF TEAK Overview

The Joint Task Force (JTF) PACTEAK is a fictional scenario used for illustrative purposes. Joint Task Force (JTF) PACTEAK is tasked to repel Kalimantan forces from Brunei and East Malaysia, degrade Kalimantan military capability, and restore territorial integrity of Brunei and East Malaysia. Following restoration of territorial integrity JTF PACTEAK forces will conduct post conflict reconstruction as necessary. Operations will be limited to East Malaysia and Brunei, except for airstrikes against military airfields in Kalimantan. Without CJTF PACTEAK approval, attacks on naval targets will be restricted to the territorial waters of East Malaysia and Brunei. Key players such as, U.S., East Malaysia, and Brunei forces should be able to collaborate across geographical boundaries depicted in Figure 3, 4 and 5. The arrows show a coordinated attack of land, sea and air forces.

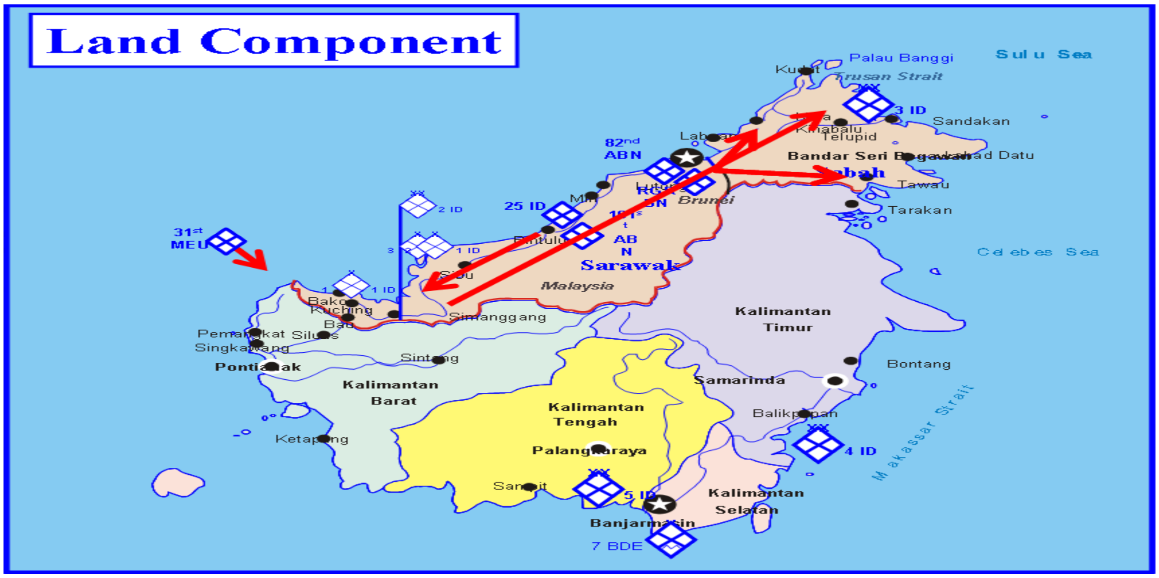


Figure 5. CFLCC (From: [17])

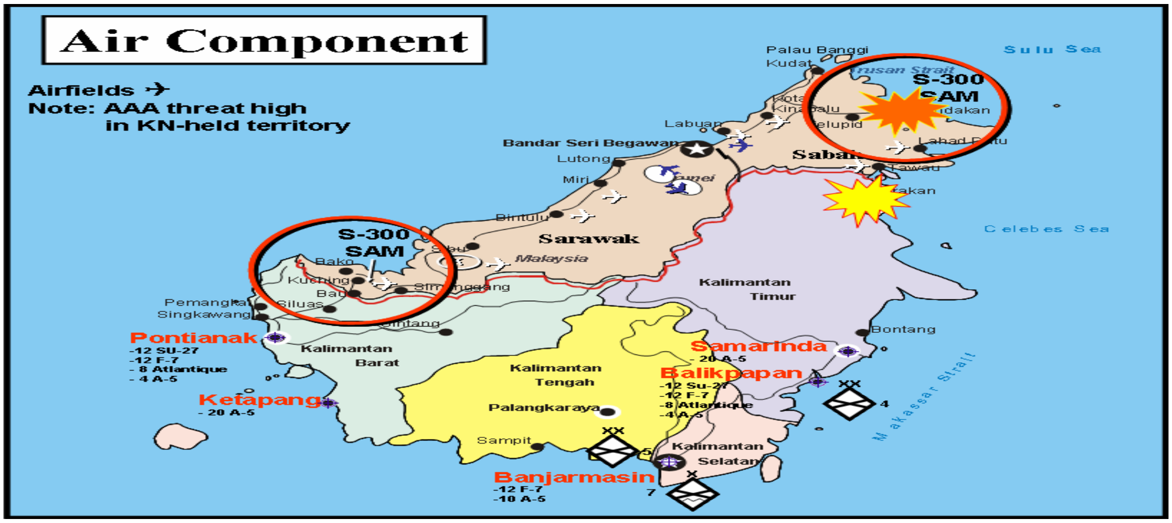


Figure 6. CFACC (From: [17])

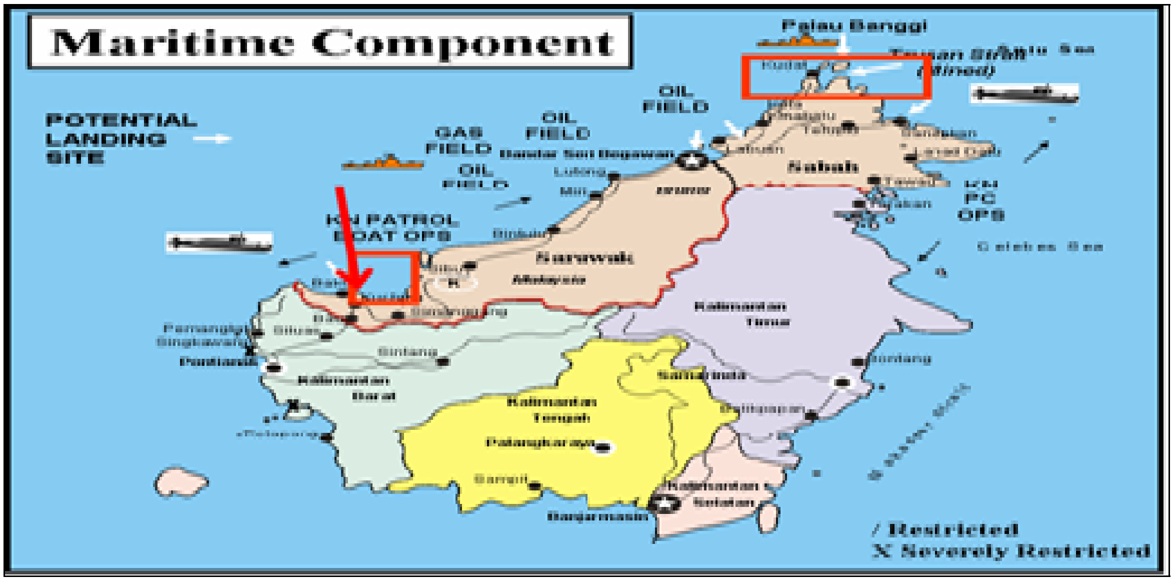


Figure 7. CFMCC (From: [17])

2. PACTEAK Wiki Implementation

In order to conduct such operations, all services and coalition forces may need to manage a complex plan that will allow each to react with real-time changes in the battlespace. Wiki technology may aid in collaboration across geographical boundaries. Therefore, JTF PACTEAK has chosen to utilize a wiki to ensure that all key players (Coalition Forces Land Component Commander

(CFLCC), Coalition Forces Maritime Component Commander (CFMCC), Coalition Forces Air Component Commander (CFACC) and host nation forces) are involved in the planning process and cultural barriers are removed to increase combat effectiveness. For the planning phase, JTF will configure the wiki for a centralized discretionary access control to maintain operational security (OPSEC). Without a detailed adherence to discretionary access controls cross international boundaries the scenario depicted above may prove to be unsuccessful in collaboration with coalition partners.

THIS PAGE INTENTIONALLY LEFT BLANK

V. EXPERIMENTATION

The discretionary access controls were tested on MediaWiki and TWiki. Both wiki engines were tested to establish a baseline and then configured to represent the four DAC controls. The experimentation method and results are presented in this section.

A. MEDIAWIKI

MediaWiki is the most recognizable wiki engine today [28]. Its most popular site, Wikipedia, receives over three millions hits per day [29]. In MediaWiki, when a user becomes a member, that user can hold four different roles. Those roles consist of *registered* users, *bot* users, *sysop* users, and *bureaucrat* users. Like that of other operating systems DAC, such as Microsoft, those roles or rights become additive as each role is upgraded. Moreover, within each role, certain control permissions are assumed. Those control permissions are *read, editing, management*, and *administration* permissions to wiki pages. Table 2 delineates each role and explains permissions associated with that role.

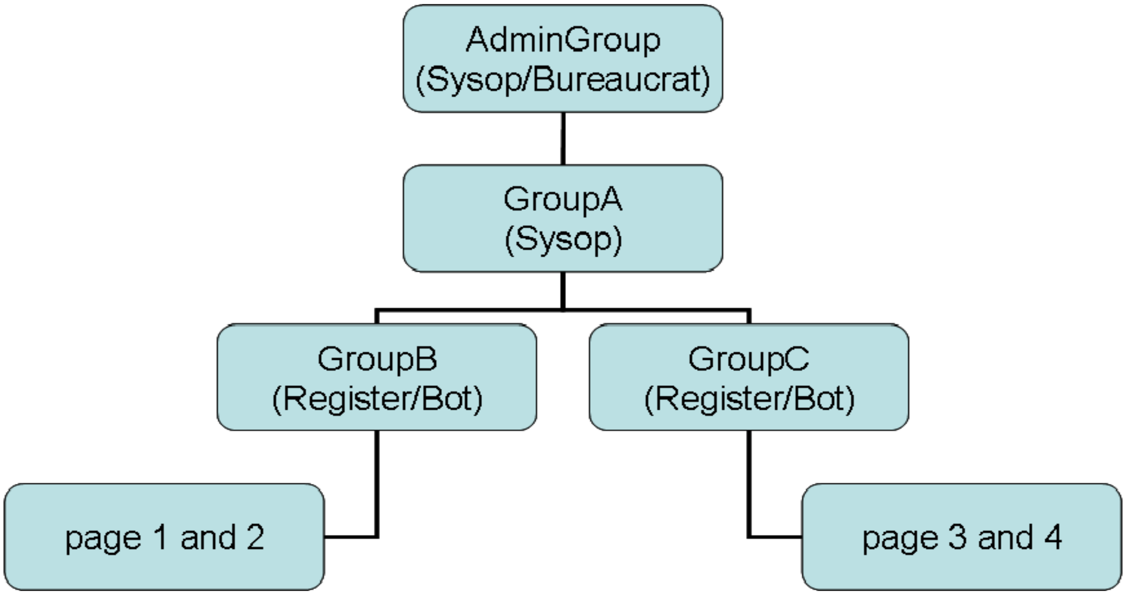
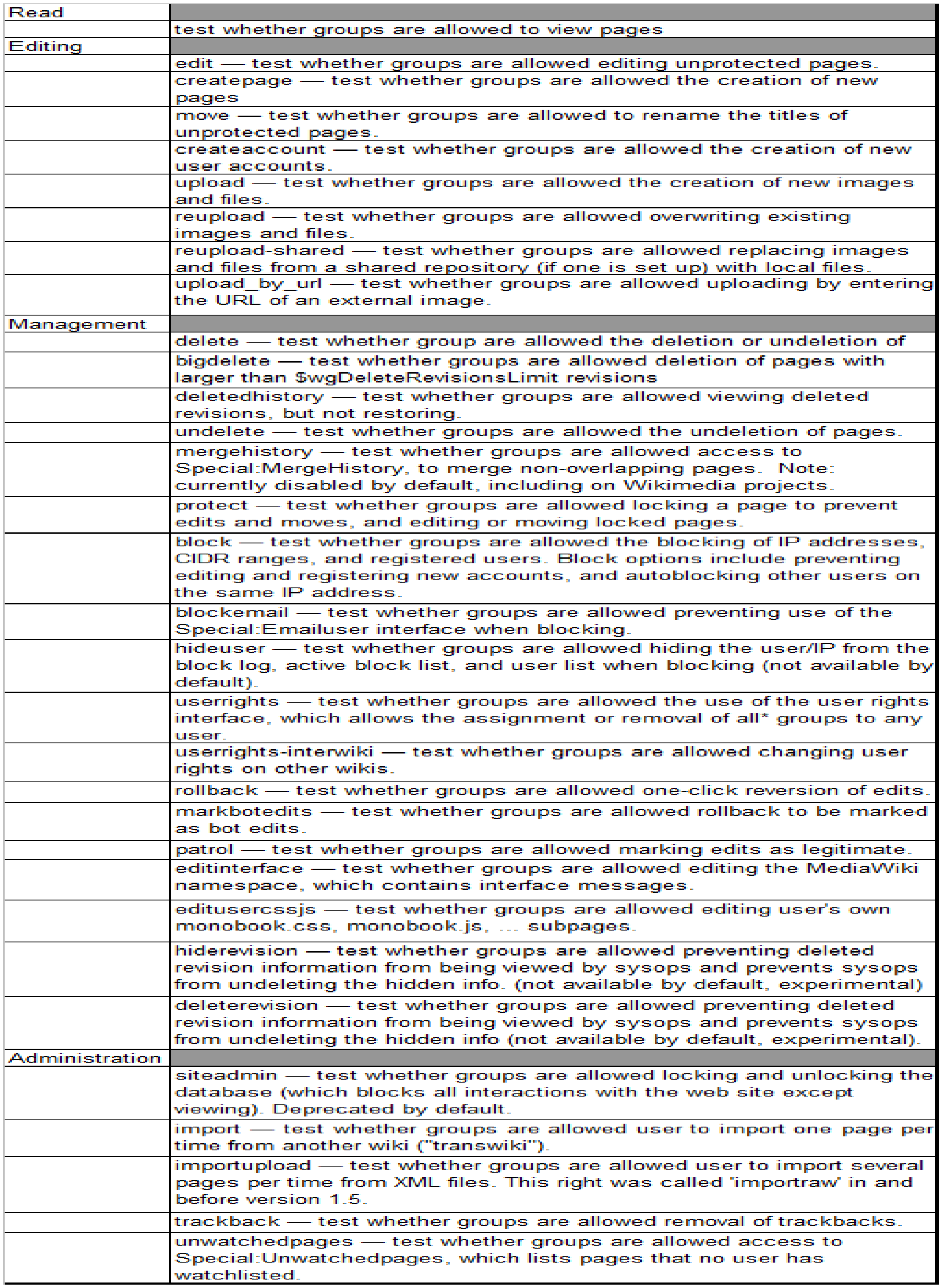


Figure 8. MediaWiki Topology

The topology used for the functional testing of MediaWiki is similar to a hierarchical organizational structure. In order to represent each DAC policy, the groups established were: groupA, groupB, and groupC. Figure 6 shows which group held what control mode. Then, MediaWiki was configured to represent the four DAC configurations policies: hierarchical, centralized, ownership, and laissez-faire. This entailed changes to the *Localsettings.php* file. Furthermore, rights were changed in the Localsettings.php file to produce the desired four policies for testing purposes. However, changes to the Localsettings.php file will require a user with sysop control permissions.

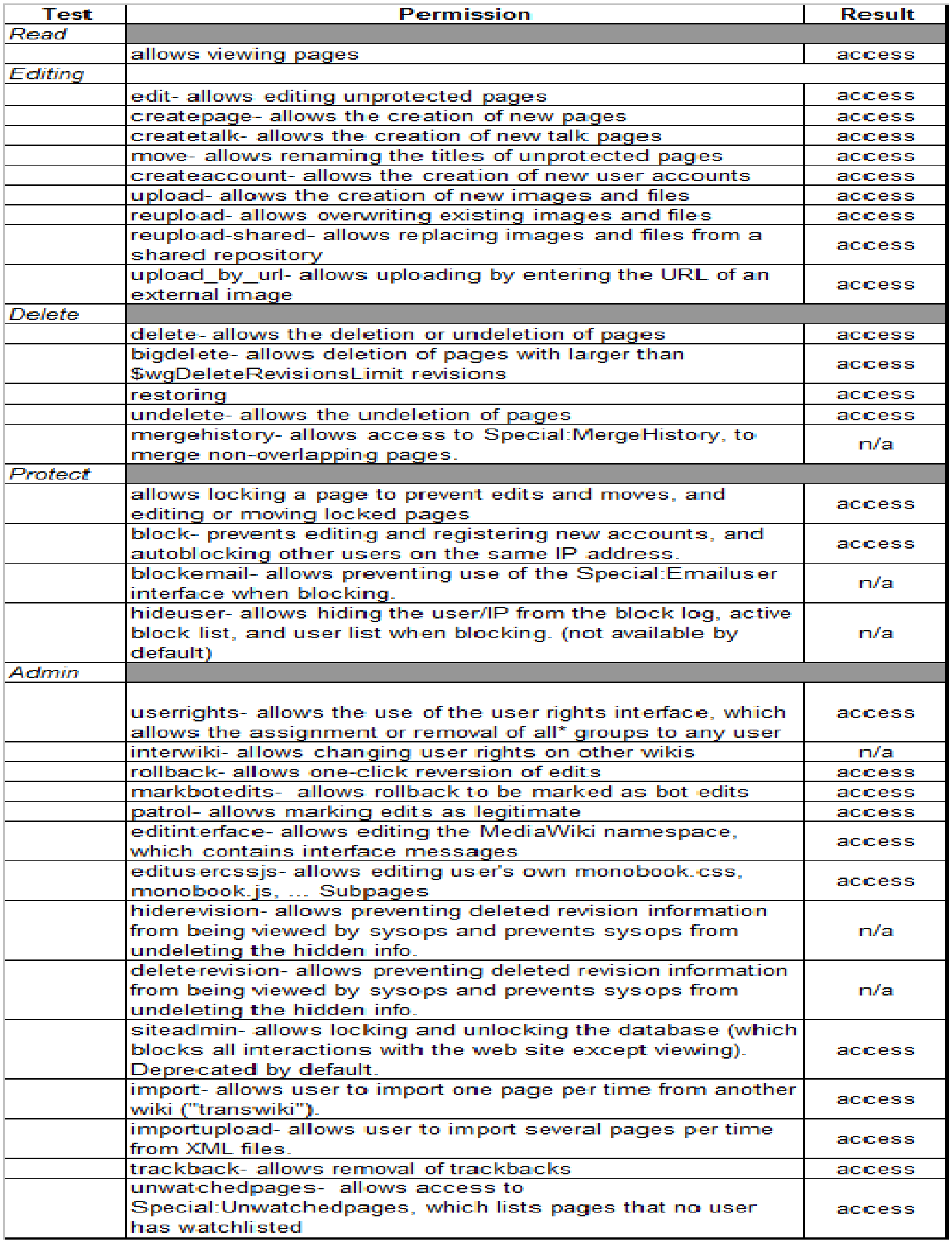
1. MediaWiki Baseline Testing

The standard setup shows how MediaWiki is set up prior to changing the Localsettings.php file. Baseline testing was conducted utilizing test plan shown in Table 2 and Table 3 show expected results from conducting the test.



Read	
	test whether groups are allowed to view pages
Editing	
	edit — test whether groups are allowed editing unprotected pages.
	createpage — test whether groups are allowed the creation of new pages
	move — test whether groups are allowed to rename the titles of unprotected pages.
	createaccount — test whether groups are allowed the creation of new user accounts.
	upload — test whether groups are allowed the creation of new images and files.
	reupload — test whether groups are allowed overwriting existing images and files.
	reupload-shared — test whether groups are allowed replacing images and files from a shared repository (if one is set up) with local files.
	upload_by_url — test whether groups are allowed uploading by entering the URL of an external image.
Management	
	delete — test whether group are allowed the deletion or undeletion of
	bigdelete — test whether groups are allowed deletion of pages with larger than $wgDeleteRevisionsLimit revisions
	deletedhistory — test whether groups are allowed viewing deleted revisions, but not restoring.
	undelete — test whether groups are allowed the undeletion of pages.
	mergehistory — test whether groups are allowed access to Special:MergeHistory, to merge non-overlapping pages. Note: currently disabled by default, including on Wikimedia projects.
	protect — test whether groups are allowed locking a page to prevent edits and moves, and editing or moving locked pages.
	block — test whether groups are allowed the blocking of IP addresses, CIDR ranges, and registered users. Block options include preventing editing and registering new accounts, and autoblocking other users on the same IP address.
	blockemail — test whether groups are allowed preventing use of the Special:Emailuser interface when blocking.
	hideuser — test whether groups are allowed hiding the user/IP from the block log, active block list, and user list when blocking (not available by default).
	userrights — test whether groups are allowed the use of the user rights interface, which allows the assignment or removal of all* groups to any user.
	userrights-interwiki — test whether groups are allowed changing user rights on other wikis.
	rollback — test whether groups are allowed one-click reversion of edits.
	markbotedits — test whether groups are allowed rollback to be marked as bot edits.
	patrol — test whether groups are allowed marking edits as legitimate.
	editinterface — test whether groups are allowed editing the MediaWiki namespace, which contains interface messages.
	editusercssjs — test whether groups are allowed editing user's own monobook.css, monobook.js, ... subpages.
	hiderevision — test whether groups are allowed preventing deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info. (not available by default, experimental)
	deleterevision — test whether groups are allowed preventing deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info (not available by default, experimental).
Administration	
	siteadmin — test whether groups are allowed locking and unlocking the database (which blocks all interactions with the web site except viewing). Deprecated by default.
	import — test whether groups are allowed user to import one page per time from another wiki ("transwiki").
	importupload — test whether groups are allowed user to import several pages per time from XML files. This right was called 'importraw' in and before version 1.5.
	trackback — test whether groups are allowed removal of trackbacks.
	unwatchedpages — test whether groups are allowed access to Special:Unwatchedpages, which lists pages that no user has watchlisted.

Table 2. MediaWiki Baseline Test Plan (From: [26])



Test	Permission	Result
Read		
	allows viewing pages	access
Editing		
	edit- allows editing unprotected pages	access
	createpage- allows the creation of new pages	access
	createtalk- allows the creation of new talk pages	access
	move- allows renaming the titles of unprotected pages	access
	createaccount- allows the creation of new user accounts	access
	upload- allows the creation of new images and files	access
	reupload- allows overwriting existing images and files	access
	reupload-shared- allows replacing images and files from a shared repository	access
	upload_by_url- allows uploading by entering the URL of an external image	access
Delete		
	delete- allows the deletion or undeletion of pages	access
	bigdelete- allows deletion of pages with larger than $wgDeleteRevisionsLimit revisions	access
	restoring	access
	undelete- allows the undeletion of pages	access
	mergehistory- allows access to Special:MergeHistory, to merge non-overlapping pages.	n/a
Protect		
	allows locking a page to prevent edits and moves, and editing or moving locked pages	access
	block- prevents editing and registering new accounts, and autoblocking other users on the same IP address.	access
	blockemail- allows preventing use of the Special:Emailuser interface when blocking.	n/a
	hideuser- allows hiding the user/IP from the block log, active block list, and user list when blocking. (not available by default)	n/a
Admin		
	userrights- allows the use of the user rights interface, which allows the assignment or removal of all* groups to any user	access
	interwiki- allows changing user rights on other wikis	n/a
	rollback- allows one-click reversion of edits	access
	markbotedits- allows rollback to be marked as bot edits	access
	patrol- allows marking edits as legitimate	access
	editinterface- allows editing the MediaWiki namespace, which contains interface messages	access
	editusercssjs- allows editing user's own monobook.css, monobook.js, ... Subpages	access
	hiderevision- allows preventing deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	deleterevision- allows preventing deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	siteadmin- allows locking and unlocking the database (which blocks all interactions with the web site except viewing). Deprecated by default.	access
	import- allows user to import one page per time from another wiki ("transwiki").	access
	importupload- allows user to import several pages per time from XML files.	access
	trackback- allows removal of trackbacks	access
	unwatchedpages- allows access to Special:Unwatchedpages, which lists pages that no user has watchlisted	access

Table 3. MediaWiki Baseline Test Results (From: [26])

2. TWiki

The basic structure of TWiki consists of *topics* and *webs.* Topics are nothing more than wiki pages. A web is a collection of related topics. In terms of file management, a topic is considered a file within a web sub-directory. Moreover, a web is considered a sub-directory within the main data directory. TWiki default settings allow registered users the ability to upload and download files from one wiki web site to another. The current default file structure is favorable to a hierarchical configuration. Each web or topic can be assigned specific permissions based upon TWiki default modes [21]. This testing topology will be similar to Microsoft Server 2003. The figure below delineates how this test topology was structured.

Configuring TWiki to represent DoD command structures (hierarchical) and files are readily available and can be set by use of the web browser interface. The same premise of groups was used: TwikiAGroup, TWikiBGroup, and TWikiCGroup were created in order to lay out the required testing architect. Additionally, by default the TWikiAdminGroup was already established for administrator level permissions. Figure 9 shows how the groups were organized for testing purposes.

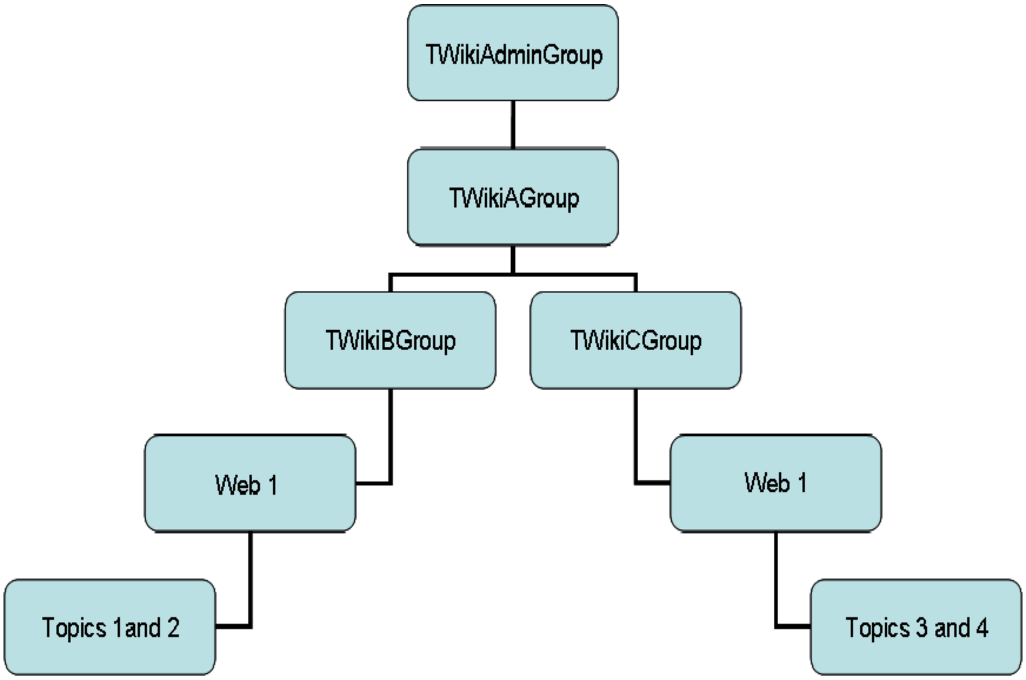
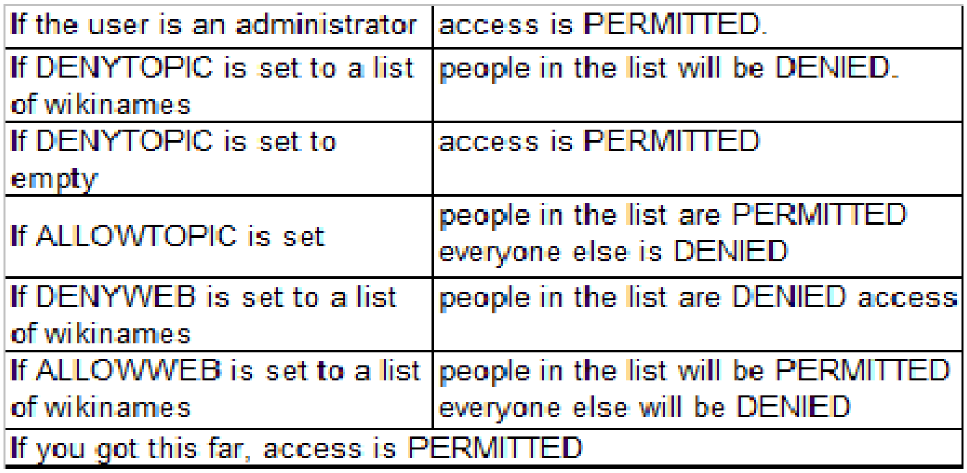


Figure 9. TWiki Topology

TWiki has two basic roles a user may be assigned, either an administrator role or a register user. An administrator has access to all permissions modes and can deny or allow other users access permission to webs or topics. However, regular uses are given default permissions to view or change topics or webs created by that user and not other users. Those roles were assigned to either TWikiAdminGroup,TWikiAGroup, TWikiBGroup, or TWikiCGroup.

There are three permission modes associated with TWiki DAC permission policy. The following modes are *View, Change,* and *Rename*. View means a user is able to view and search wiki content. Change means a user is allowed create new topics, change topics or attach files [23]. Rename allows a user to rename topics within a web. The rename mode is restricted to the TWikiAdminGroup by default. When deciding whether to grant access, TWiki evaluates the following rules in order (read from the top of the list; if the logic arrives at PERMITTED or DENIED, that applies immediately and no more rules are applied). Bear in mind that VIEW, CHANGE and RENAME access may be granted/denied separately. TWiki modes were applied to each group as outlined below in Table 4 [23].

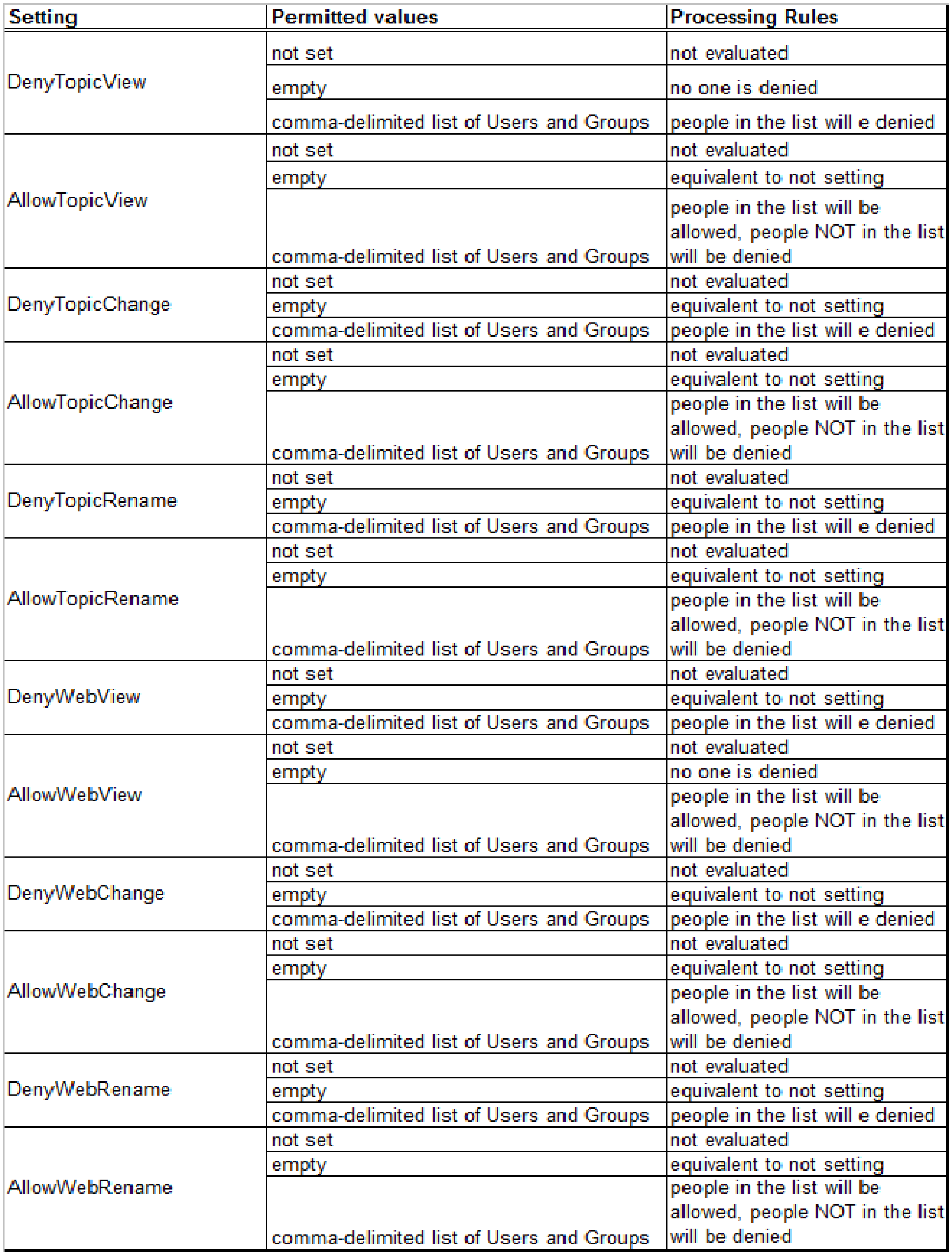


If the user is an administrator	access is PERMITTED.
If DENYTOPIC is set to a list of wikinames	people in the list will be DENIED.
If DENYTOPIC is set to empty	access is PERMITTED
If ALLOWTOPIC is set	people in the list are PERMITTED everyone else is DENIED
If DENYWEB is set to a list of wikinames	people in the list are DENIED access
If ALLOWWEB is set to a list of wikinames	people in the list will be PERMITTED everyone else will be DENIED
If you got this far, access is PERMITTED	

Table 4. TWiki Modes (From: [23])

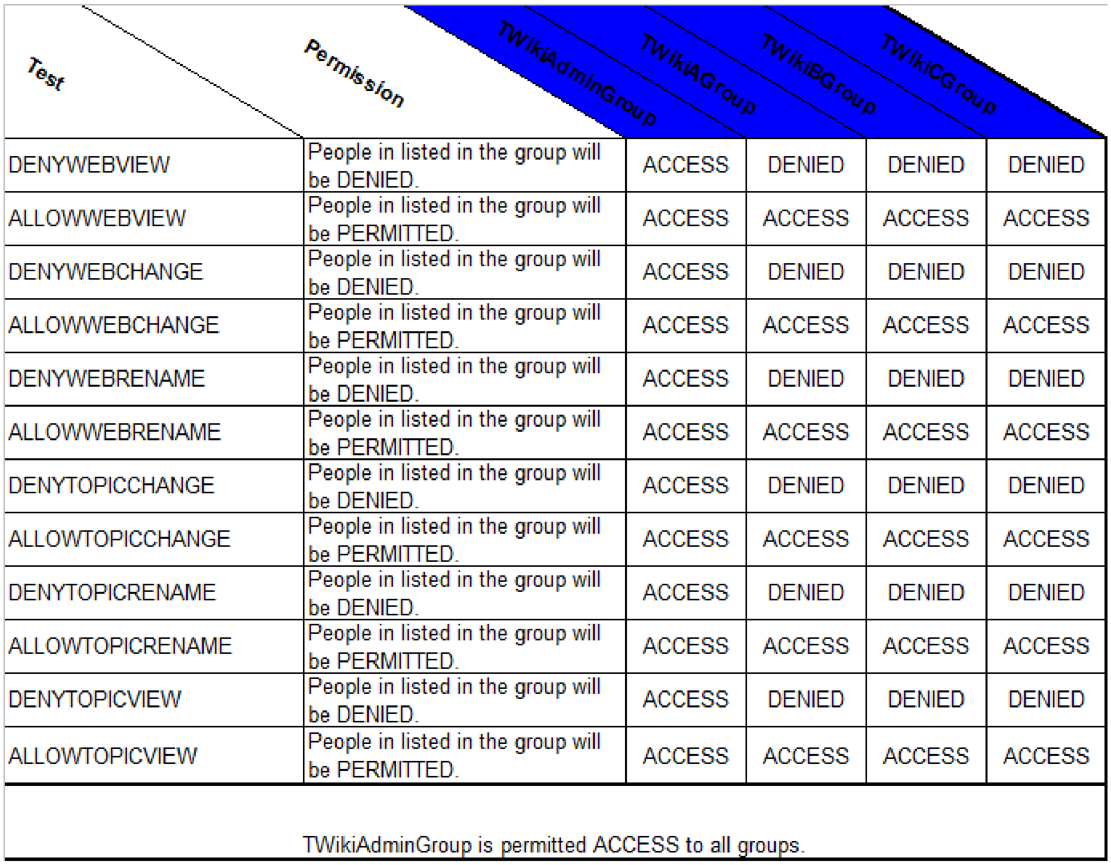
a. *TWiki Baseline Testing*

Table 5 shows the baseline test plan control modes and how those controls are delineated. By default, the only group available is the TWikiAdminGroup. Therefore, three other groups had to be created and given registered users' permission rights: AllowTopicView, AllowTopicChange, AllowTopicRename, AllowWebView AllowWebChange and AllowWebRename. A baseline test was also conducted and the results are shown in Table 6.



Setting	Permitted values	Processing Rules
DenyTopicView	not set	not evaluated
	empty	no one is denied
	comma-delimited list of Users and Groups	people in the list will e denied
AllowTopicView	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will be allowed, people NOT in the list will be denied
DenyTopicChange	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will e denied
AllowTopicChange	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will be allowed, people NOT in the list will be denied
DenyTopicRename	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will e denied
AllowTopicRename	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will be allowed, people NOT in the list will be denied
DenyWebView	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will e denied
AllowWebView	not set	not evaluated
	empty	no one is denied
	comma-delimited list of Users and Groups	people in the list will be allowed, people NOT in the list will be denied
DenyWebChange	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will e denied
AllowWebChange	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will be allowed, people NOT in the list will be denied
DenyWebRename	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will e denied
AllowWebRename	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the list will be allowed, people NOT in the list will be denied

Table 5. TWiki Baseline Test Plan (From: [23])



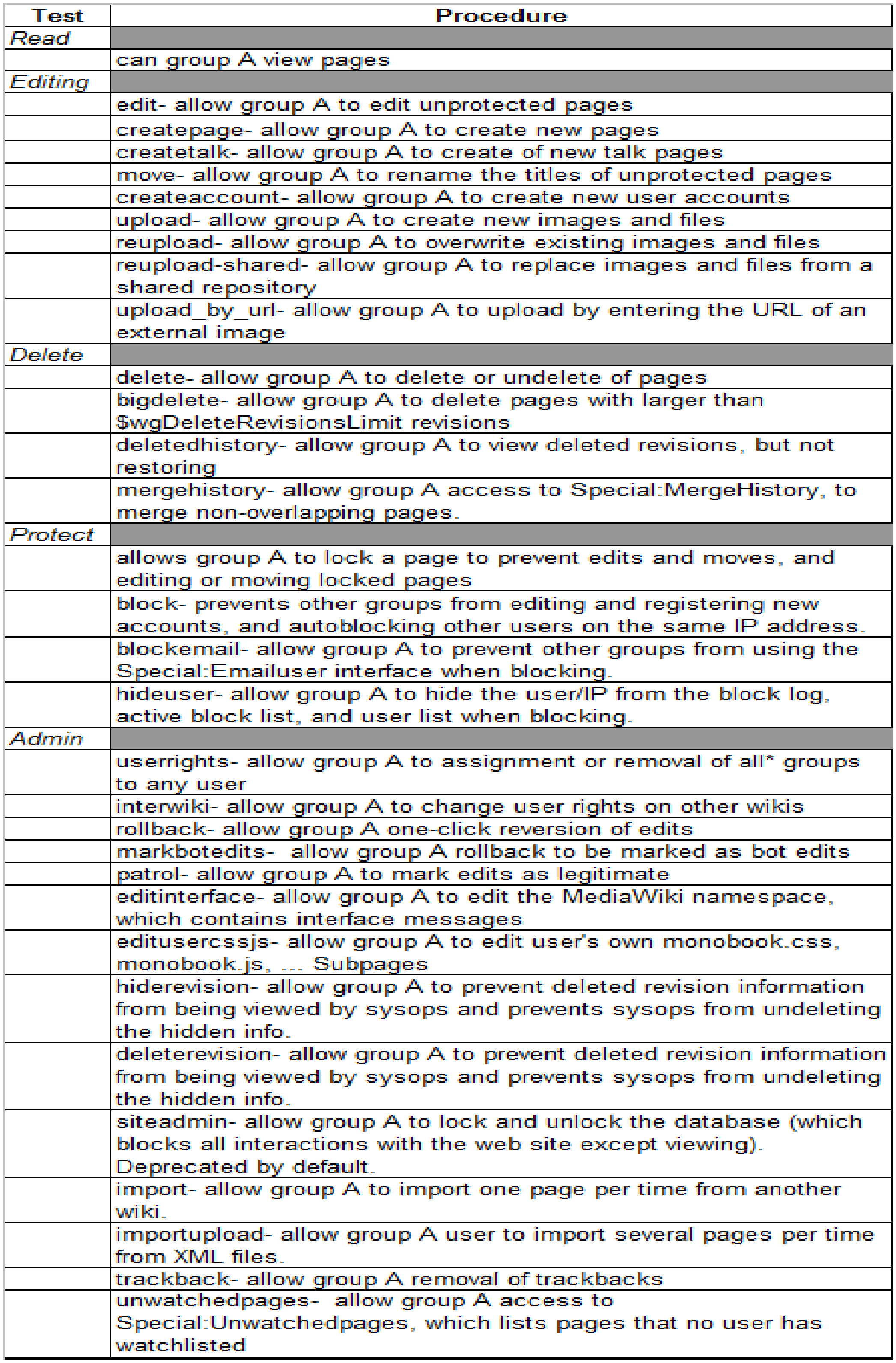
Test	Permission	TWikiAdminGroup	TWikiAGroup	TWikiBGroup	TWikiCGroup
DENYWEBVIEW	People in listed in the group will be DENIED.	ACCESS	DENIED	DENIED	DENIED
ALLOWWEBVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYWEBCHANGE	People in listed in the group will be DENIED.	ACCESS	DENIED	DENIED	DENIED
ALLOWWEBCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYWEBRENAME	People in listed in the group will be DENIED.	ACCESS	DENIED	DENIED	DENIED
ALLOWWEBRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYTOPICCHANGE	People in listed in the group will be DENIED.	ACCESS	DENIED	DENIED	DENIED
ALLOWTOPICCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYTOPICRENAME	People in listed in the group will be DENIED.	ACCESS	DENIED	DENIED	DENIED
ALLOWTOPICRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYTOPICVIEW	People in listed in the group will be DENIED.	ACCESS	DENIED	DENIED	DENIED
ALLOWTOPICVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
TWikiAdminGroup is permitted ACCESS to all groups.					

Table 6. TWiki Baseline Results (From: [23])

3. MediaWiki Discretionary Access Control Testing

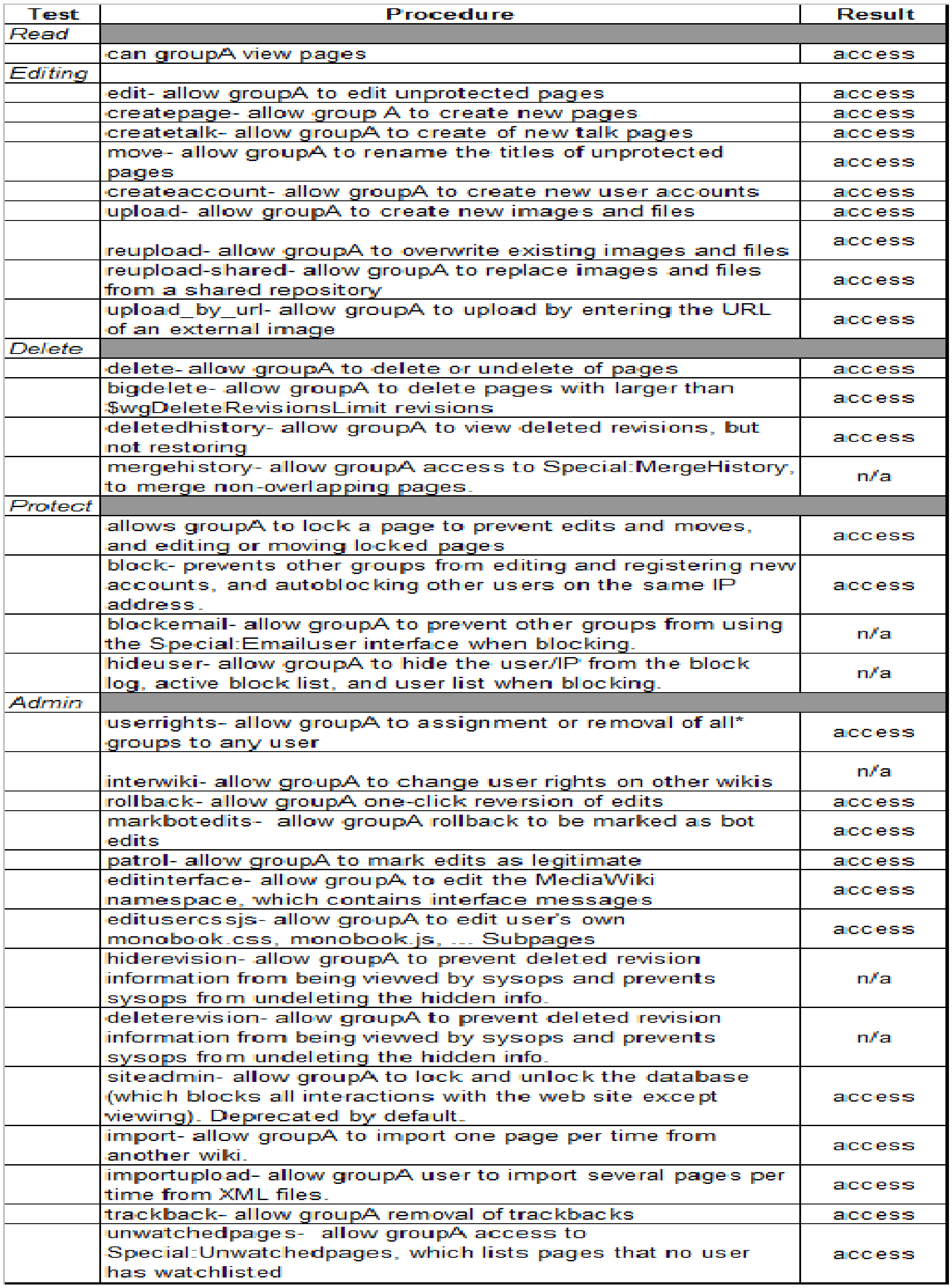
a. MediaWiki Hierarchical Control

Rights were incorporate under one administrative group for testing, management and administrative control. The rationale was that an administrator in most organizations normally holds rights held by management control. In order to establish a hierarchical policy with the use of groups, groupA was given sysop and the bureaucrat role and the control permissions associated with those roles. This, however, would give groupA read, edit, delete, protect, and administrative control permissions with the ability to change users rights. GroupB was given the sysops role only, which entails the ability to read, edit, delete, protect, and administrative control permissions with the exception of user rights control permission. GroupC was given register users, which possess' the ability to read and edit, with the exception of unprotect and bigdelete (right to delete large files) control permissions. GroupA could delegate rights down the organizational structure by changing the code to represent True or False. In addition, group B could delegate down, the chain of command to groupC. In order to achieve a hierarchical control structure the code in the Localsettings.php file was changed to reflect the desired DAC policy; Table 7 shows the test plan and Table 8 reflects the given results.



Test	Procedure
Read	
	can group A view pages
Editing	
	edit- allow group A to edit unprotected pages
	createpage- allow group A to create new pages
	createtalk- allow group A to create of new talk pages
	move- allow group A to rename the titles of unprotected pages
	createaccount- allow group A to create new user accounts
	upload- allow group A to create new images and files
	reupload- allow group A to overwrite existing images and files
	reupload-shared- allow group A to replace images and files from a shared repository
	upload_by_url- allow group A to upload by entering the URL of an external image
Delete	
	delete- allow group A to delete or undelete of pages
	bigdelete- allow group A to delete pages with larger than $wgDeleteRevisionsLimit revisions
	deletedhistory- allow group A to view deleted revisions, but not restoring
	mergehistory- allow group A access to Special:MergeHistory, to merge non-overlapping pages.
Protect	
	allows group A to lock a page to prevent edits and moves, and editing or moving locked pages
	block- prevents other groups from editing and registering new accounts, and autoblocking other users on the same IP address.
	blockemail- allow group A to prevent other groups from using the Special:Emailuser interface when blocking.
	hideuser- allow group A to hide the user/IP from the block log, active block list, and user list when blocking.
Admin	
	userrights- allow group A to assignment or removal of all* groups to any user
	interwiki- allow group A to change user rights on other wikis
	rollback- allow group A one-click reversion of edits
	markbotedits- allow group A rollback to be marked as bot edits
	patrol- allow group A to mark edits as legitimate
	editinterface- allow group A to edit the MediaWiki namespace, which contains interface messages
	editusercssjs- allow group A to edit user's own monobook.css, monobook.js, ... Subpages
	hiderevision- allow group A to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	deleterevision- allow group A to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	siteadmin- allow group A to lock and unlock the database (which blocks all interactions with the web site except viewing). Deprecated by default.
	import- allow group A to import one page per time from another wiki.
	importupload- allow group A user to import several pages per time from XML files.
	trackback- allow group A removal of trackbacks
	unwatchedpages- allow group A access to Special:Unwatchedpages, which lists pages that no user has watchlisted

Table 7. MediaWiki Hierarchical Test Plan (From: [26])

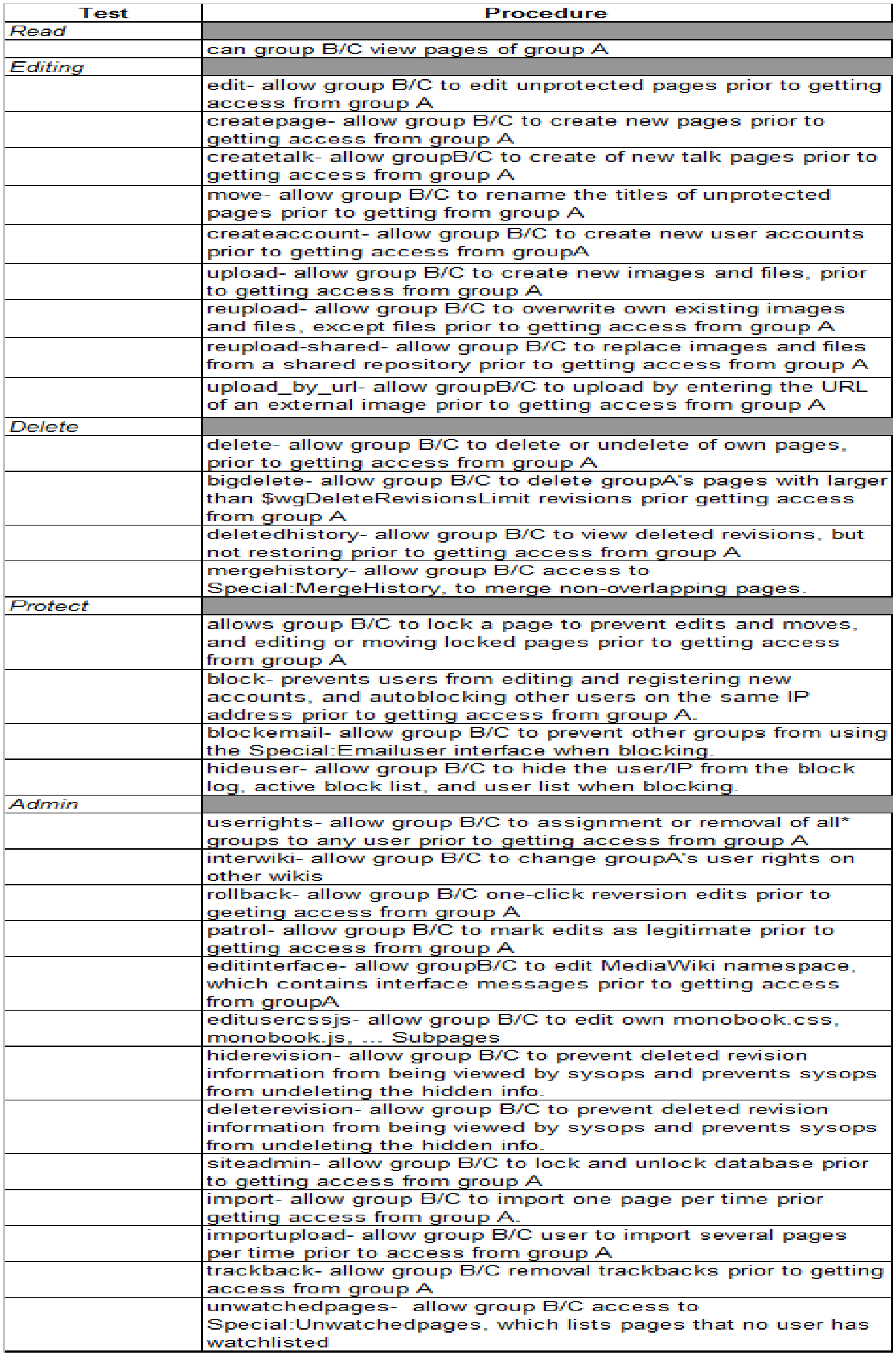


Test	Procedure	Result
Read		
	can groupA view pages	access
Editing		
	edit- allow groupA to edit unprotected pages	access
	createpage- allow group A to create new pages	access
	createtalk- allow groupA to create of new talk pages	access
	move- allow groupA to rename the titles of unprotected pages	access
	createaccount- allow groupA to create new user accounts	access
	upload- allow groupA to create new images and files	access
	reupload- allow groupA to overwrite existing images and files	access
	reupload-shared- allow groupA to replace images and files from a shared repository	access
	upload_by_url- allow groupA to upload by entering the URL of an external image	access
Delete		
	delete- allow groupA to delete or undelete of pages	access
	bigdelete- allow groupA to delete pages with larger than $wgDeleteRevisionsLimit revisions	access
	deletedhistory- allow groupA to view deleted revisions, but not restoring	access
	mergehistory- allow groupA access to Special:MergeHistory, to merge non-overlapping pages.	n/a
Protect		
	allows groupA to lock a page to prevent edits and moves, and editing or moving locked pages	access
	block- prevents other groups from editing and registering new accounts, and autoblocking other users on the same IP address.	access
	blockemail- allow groupA to prevent other groups from using the Special:Emailuser interface when blocking.	n/a
	hideuser- allow groupA to hide the user/IP from the block log, active block list, and user list when blocking.	n/a
Admin		
	userrights- allow groupA to assignment or removal of all* groups to any user	access
	interwiki- allow groupA to change user rights on other wikis	n/a
	rollback- allow groupA one-click reversion of edits	access
	markbotedits- allow groupA rollback to be marked as bot edits	access
	patrol- allow groupA to mark edits as legitimate	access
	editinterface- allow groupA to edit the MediaWiki namespace, which contains interface messages	access
	editusercssjs- allow groupA to edit user's own monobook.css, monobook.js, ... Subpages	access
	hiderevision- allow groupA to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	deleterevision- allow groupA to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	siteadmin- allow groupA to lock and unlock the database (which blocks all interactions with the web site except viewing). Deprecated by default.	access
	import- allow groupA to import one page per time from another wiki.	access
	importupload- allow groupA user to import several pages per time from XML files.	access
	trackback- allow groupA removal of trackbacks	access
	unwatchedpages- allow groupA access to Special:Unwatchedpages, which lists pages that no user has watchlisted	access

Table 8. MediaWiki Hierarchical Test Results (From: [26])

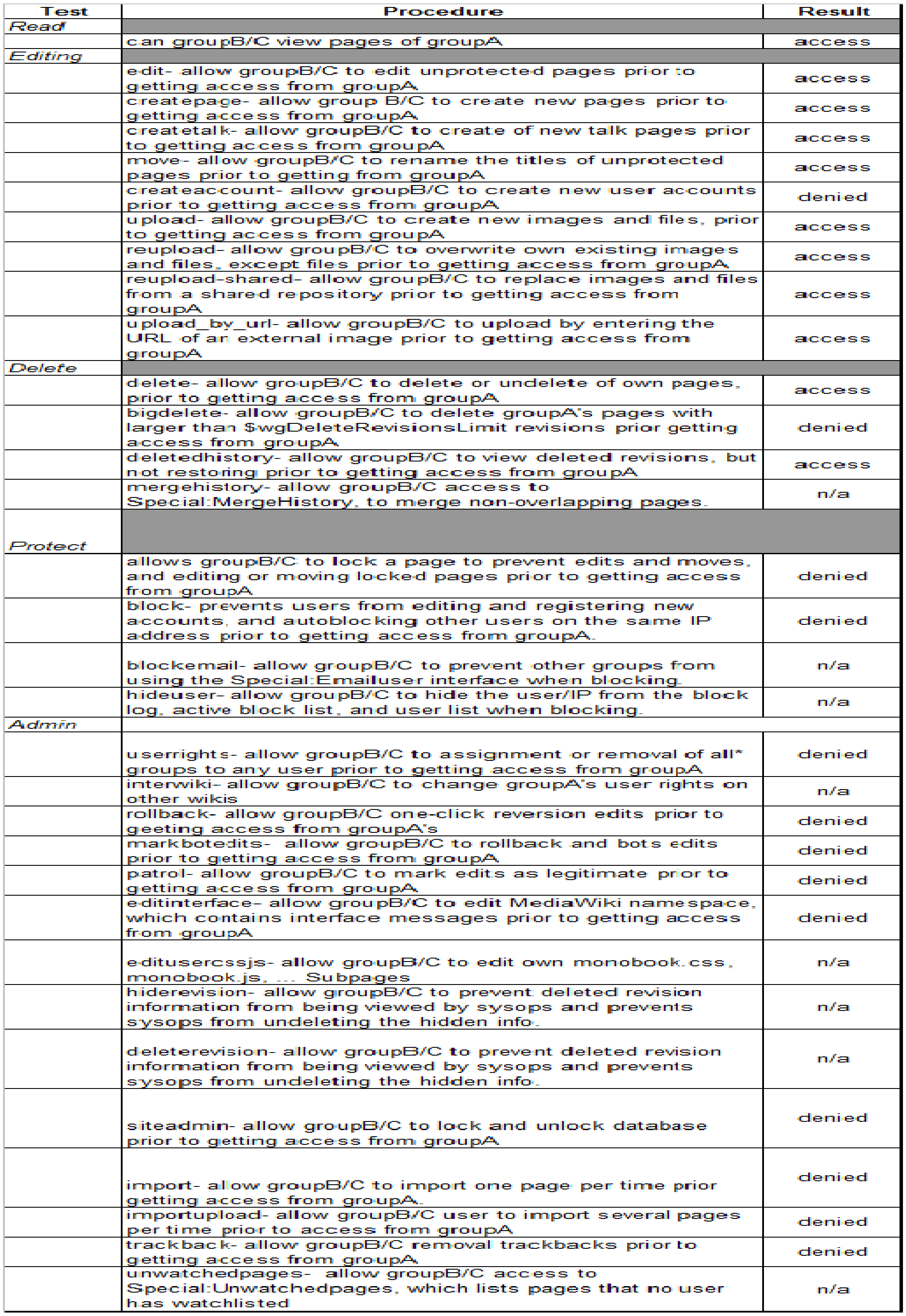
b. MediaWiki Centralized Control

The centralized permissions are similar to hierarchical control with the exception of not having the ability to delegate control to other users. However, groupB and groupC can not gain access to pages other than the right to read without prior permissions granted from groupA. The results shown below are prior to being granted access from groupA. Moreover, in order to become a new user, users would have to request to be added by groupA. GroupB and groupC were only given the right to read. Table 9 shows the test plan along with Table 10, showing test results.



Test	Procedure
Read	
	can group B/C view pages of group A
Editing	
	edit- allow group B/C to edit unprotected pages prior to getting access from group A
	createpage- allow group B/C to create new pages prior to getting access from group A
	createtalk- allow groupB/C to create of new talk pages prior to getting access from group A
	move- allow group B/C to rename the titles of unprotected pages prior to getting from group A
	createaccount- allow group B/C to create new user accounts prior to getting access from groupA
	upload- allow group B/C to create new images and files, prior to getting access from group A
	reupload- allow group B/C to overwrite own existing images and files, except files prior to getting access from group A
	reupload-shared- allow group B/C to replace images and files from a shared repository prior to getting access from group A
	upload_by_url- allow groupB/C to upload by entering the URL of an external image prior to getting access from group A
Delete	
	delete- allow group B/C to delete or undelete of own pages, prior to getting access from group A
	bigdelete- allow group B/C to delete groupA's pages with larger than $wgDeleteRevisionsLimit revisions prior getting access from group A
	deletedhistory- allow group B/C to view deleted revisions, but not restoring prior to getting access from group A
	mergehistory- allow group B/C access to Special:MergeHistory, to merge non-overlapping pages.
Protect	
	allows group B/C to lock a page to prevent edits and moves, and editing or moving locked pages prior to getting access from group A
	block- prevents users from editing and registering new accounts, and autoblocking other users on the same IP address prior to getting access from group A.
	blockemail- allow group B/C to prevent other groups from using the Special:Emailuser interface when blocking.
	hideuser- allow group B/C to hide the user/IP from the block log, active block list, and user list when blocking.
Admin	
	userrights- allow group B/C to assignment or removal of all* groups to any user prior to getting access from group A
	interwiki- allow group B/C to change groupA's user rights on other wikis
	rollback- allow group B/C one-click reversion edits prior to geeting access from group A
	patrol- allow group B/C to mark edits as legitimate prior to getting access from group A
	editinterface- allow groupB/C to edit MediaWiki namespace, which contains interface messages prior to getting access from groupA
	editusercssjs- allow group B/C to edit own monobook.css, monobook.js, ... Subpages
	hiderevision- allow group B/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	deleterevision- allow group B/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	siteadmin- allow group B/C to lock and unlock database prior to getting access from group A
	import- allow group B/C to import one page per time prior getting access from group A.
	importupload- allow group B/C user to import several pages per time prior to access from group A
	trackback- allow group B/C removal trackbacks prior to getting access from group A
	unwatchedpages- allow group B/C access to Special:Unwatchedpages, which lists pages that no user has watchlisted

Table 9. MediaWiki Centralized Test Plan (From: [26])

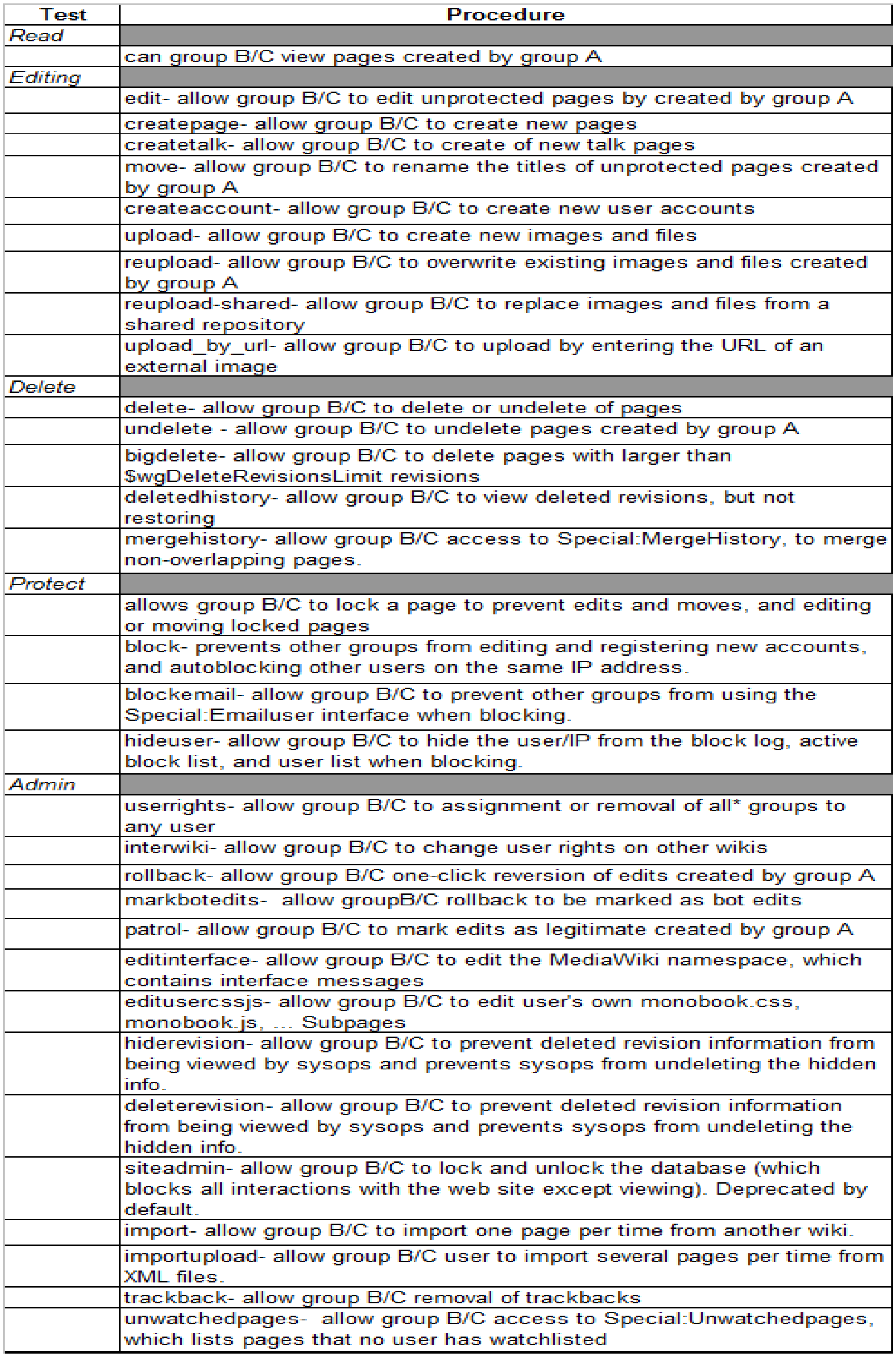


Test	Procedure	Result
Read		
	can groupB/C view pages of groupA	access
Editing		
	edit- allow groupB/C to edit unprotected pages prior to getting access from groupA	access
	createpage- allow group B/C to create new pages prior to getting access from groupA	access
	createtalk- allow groupB/C to create of new talk pages prior to getting access from groupA	access
	move- allow groupB/C to rename the titles of unprotected pages prior to getting from groupA	access
	createaccount- allow groupB/C to create new user accounts prior to getting access from groupA	denied
	upload- allow groupB/C to create new images and files, prior to getting access from groupA	access
	reupload- allow groupB/C to overwrite own existing images and files, except files prior to getting access from groupA	access
	reupload-shared- allow groupB/C to replace images and files from a shared repository prior to getting access from groupA	access
	upload_by_url- allow groupB/C to upload by entering the URL of an external image prior to getting access from groupA	access
Delete		
	delete- allow groupB/C to delete or undelete of own pages, prior to getting access from groupA	access
	bigdelete- allow groupB/C to delete groupA's pages with larger than $wgDeleteRevisionsLimit revisions prior getting access from groupA	denied
	deletedhistory- allow groupB/C to view deleted revisions, but not restoring prior to getting access from groupA	access
	mergehistory- allow groupB/C access to Special:MergeHistory, to merge non-overlapping pages.	n/a
Protect		
	allows groupB/C to lock a page to prevent edits and moves, and editing or moving locked pages prior to getting access from groupA	denied
	block- prevents users from editing and registering new accounts, and autoblocking other users on the same IP address prior to getting access from groupA.	denied
	blockemail- allow groupB/C to prevent other groups from using the Special:Emailuser interface when blocking.	n/a
	hideuser- allow groupB/C to hide the user/IP from the block log, active block list, and user list when blocking.	n/a
Admin		
	userrights- allow groupB/C to assignment or removal of all* groups to any user prior to getting access from groupA	denied
	interwiki- allow groupB/C to change groupA's user rights on other wikis	n/a
	rollback- allow groupB/C one-click reversion edits prior to geeting access from groupA's	denied
	markbotedits- allow groupB/C to rollback and bots edits prior to getting access from groupA	denied
	patrol- allow groupB/C to mark edits as legitimate prior to getting access from groupA	denied
	editinterface- allow groupB/C to edit MediaWiki namespace, which contains interface messages prior to getting access from groupA	denied
	editusercssjs- allow groupB/C to edit own monobook.css, monobook.js, ... Subpages	n/a
	hiderevision- allow groupB/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	deleterevision- allow groupB/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	siteadmin- allow groupB/C to lock and unlock database prior to getting access from groupA	denied
	import- allow groupB/C to import one page per time prior getting access from groupA.	denied
	importupload- allow groupB/C user to import several pages per time prior to access from groupA	denied
	trackback- allow groupB/C removal trackbacks prior to getting access from groupA	denied
	unwatchedpages- allow groupB/C access to Special:Unwatchedpages, which lists pages that no user has watchlisted	n/a

Table 10. MediaWiki Centralized Test Results (From: [26])

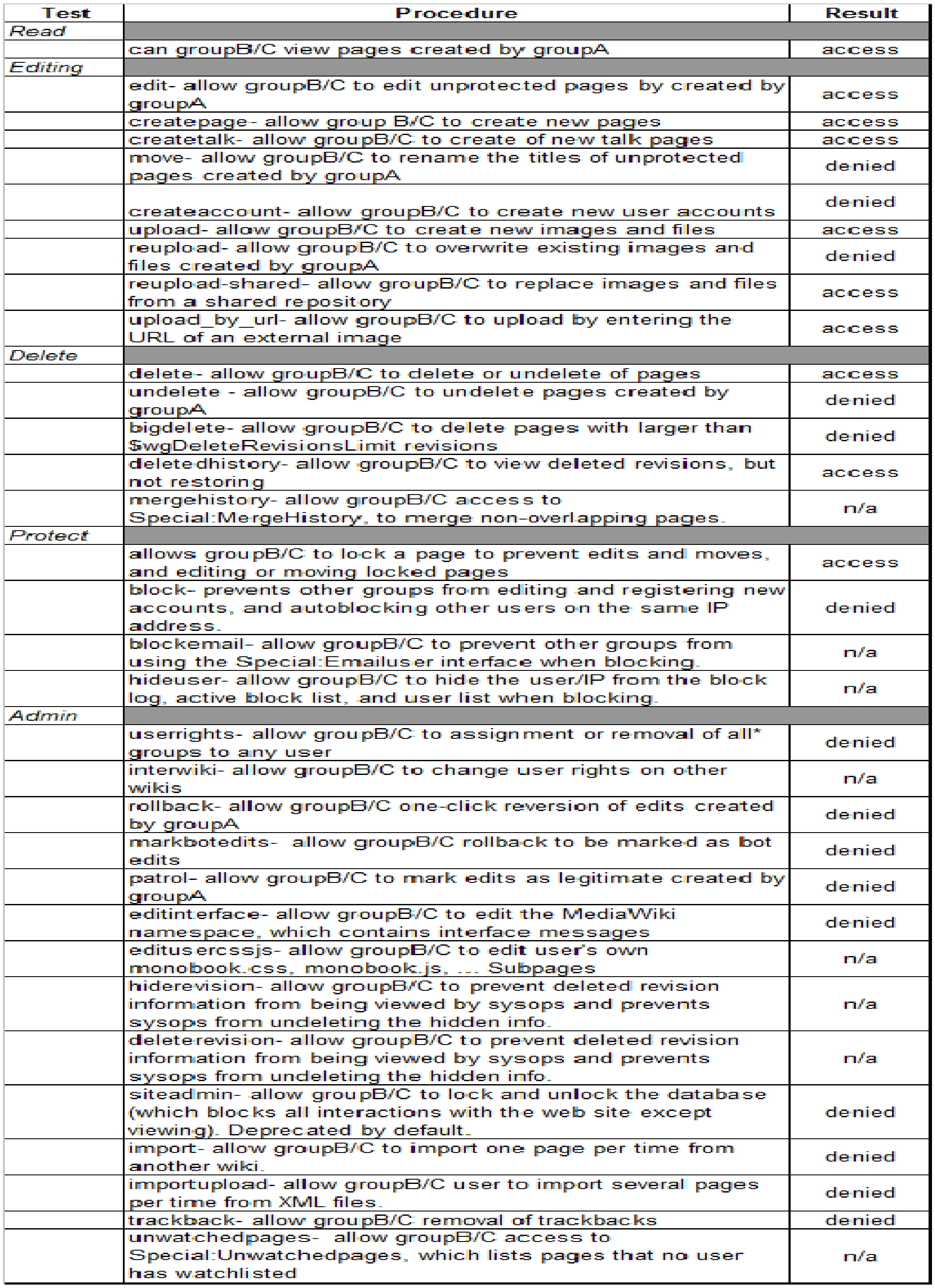
c. MediaWiki Ownership Control

The rationale for configuring ownership control permissions is to establish three test groups: groupA, groupB and groupC. GroupA was given ownership control, along with permissions to read, editing, delete, protect and administrative control permissions, with the exception of user rights control. GroupB and groupC were given register user control permissions: read, limited edit, protect, and delete rights. In other words, groupB and groupC cannot edit, lock pages or undelete pages belonging to groupA, but only reserve the rights to edit, protect and delete pages to which they have access. The test plan in Table 11 is shown below, and Table 12 shows the test results.



Test	Procedure
Read	
	can group B/C view pages created by group A
Editing	
	edit- allow group B/C to edit unprotected pages by created by group A
	createpage- allow group B/C to create new pages
	createtalk- allow group B/C to create of new talk pages
	move- allow group B/C to rename the titles of unprotected pages created by group A
	createaccount- allow group B/C to create new user accounts
	upload- allow group B/C to create new images and files
	reupload- allow group B/C to overwrite existing images and files created by group A
	reupload-shared- allow group B/C to replace images and files from a shared repository
	upload_by_url- allow group B/C to upload by entering the URL of an external image
Delete	
	delete- allow group B/C to delete or undelete of pages
	undelete - allow group B/C to undelete pages created by group A
	bigdelete- allow group B/C to delete pages with larger than $wgDeleteRevisionsLimit revisions
	deletedhistory- allow group B/C to view deleted revisions, but not restoring
	mergehistory- allow group B/C access to Special:MergeHistory, to merge non-overlapping pages.
Protect	
	allows group B/C to lock a page to prevent edits and moves, and editing or moving locked pages
	block- prevents other groups from editing and registering new accounts, and autoblocking other users on the same IP address.
	blockemail- allow group B/C to prevent other groups from using the Special:Emailuser interface when blocking.
	hideuser- allow group B/C to hide the user/IP from the block log, active block list, and user list when blocking.
Admin	
	userrights- allow group B/C to assignment or removal of all* groups to any user
	interwiki- allow group B/C to change user rights on other wikis
	rollback- allow group B/C one-click reversion of edits created by group A
	markbotedits- allow groupB/C rollback to be marked as bot edits
	patrol- allow group B/C to mark edits as legitimate created by group A
	editinterface- allow group B/C to edit the MediaWiki namespace, which contains interface messages
	editusercssjs- allow group B/C to edit user's own monobook.css, monobook.js, ... Subpages
	hiderevision- allow group B/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	deleterevision- allow group B/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	siteadmin- allow group B/C to lock and unlock the database (which blocks all interactions with the web site except viewing). Deprecated by default.
	import- allow group B/C to import one page per time from another wiki.
	importupload- allow group B/C user to import several pages per time from XML files.
	trackback- allow group B/C removal of trackbacks
	unwatchedpages- allow group B/C access to Special:Unwatchedpages, which lists pages that no user has watchlisted

Table 11. MediaWiki Ownership Test Plan (From: [26])

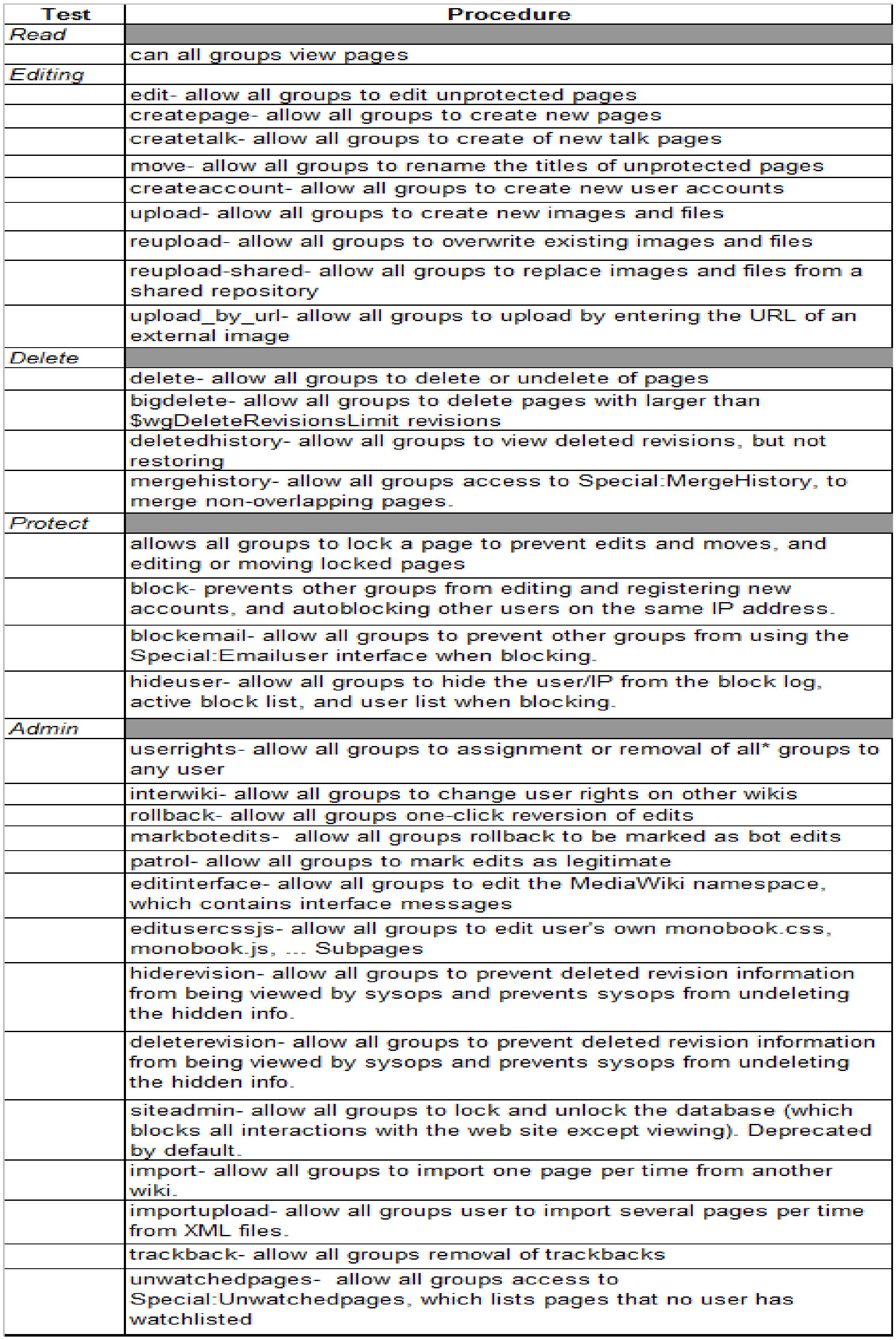


Test	Procedure	Result
Read		
	can groupB/C view pages created by groupA	access
Editing		
	edit- allow groupB/C to edit unprotected pages by created by groupA	access
	createpage- allow group B/C to create new pages	access
	createtalk- allow groupB/C to create of new talk pages	access
	move- allow groupB/C to rename the titles of unprotected pages created by groupA	denied
	createaccount- allow groupB/C to create new user accounts	denied
	upload- allow groupB/C to create new images and files	access
	reupload- allow groupB/C to overwrite existing images and files created by groupA	denied
	reupload-shared- allow groupB/C to replace images and files from a shared repository	access
	upload_by_url- allow groupB/C to upload by entering the URL of an external image	access
Delete		
	delete- allow groupB/C to delete or undelete of pages	access
	undelete - allow groupB/C to undelete pages created by groupA	denied
	bigdelete- allow groupB/C to delete pages with larger than $wgDeleteRevisionsLimit revisions	denied
	deletedhistory- allow groupB/C to view deleted revisions, but not restoring	access
	mergehistory- allow groupB/C access to Special:MergeHistory, to merge non-overlapping pages.	n/a
Protect		
	allows groupB/C to lock a page to prevent edits and moves, and editing or moving locked pages	access
	block- prevents other groups from editing and registering new accounts, and autoblocking other users on the same IP address.	denied
	blockemail- allow groupB/C to prevent other groups from using the Special:Emailuser interface when blocking.	n/a
	hideuser- allow groupB/C to hide the user/IP from the block log, active block list, and user list when blocking.	n/a
Admin		
	userrights- allow groupB/C to assignment or removal of all* groups to any user	denied
	interwiki- allow groupB/C to change user rights on other wikis	n/a
	rollback- allow groupB/C one-click reversion of edits created by groupA	denied
	markbotedits- allow groupB/C rollback to be marked as bot edits	denied
	patrol- allow groupB/C to mark edits as legitimate created by groupA	denied
	editinterface- allow groupB/C to edit the MediaWiki namespace, which contains interface messages	denied
	editusercssjs- allow groupB/C to edit user's own monobook.css, monobook.js, ... Subpages	n/a
	hiderevision- allow groupB/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	deleterevision- allow groupB/C to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	siteadmin- allow groupB/C to lock and unlock the database (which blocks all interactions with the web site except viewing). Deprecated by default.	denied
	import- allow groupB/C to import one page per time from another wiki.	denied
	importupload- allow groupB/C user to import several pages per time from XML files.	denied
	trackback- allow groupB/C removal of trackbacks	denied
	unwatchedpages- allow groupB/C access to Special:Unwatchedpages, which lists pages that no user has watchlisted	n/a

Table 12. MediaWiki Ownership Test Results (From: [26])

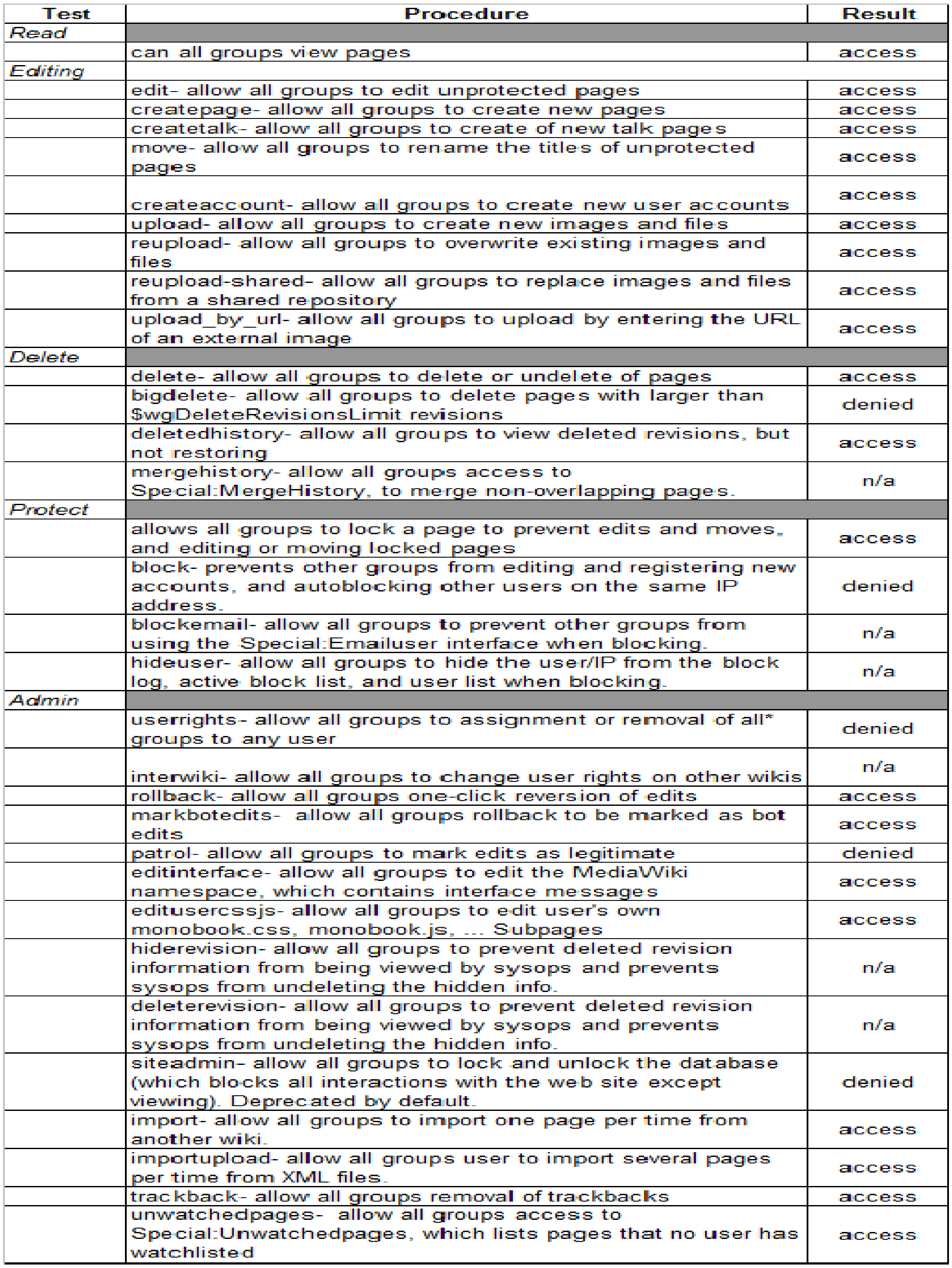
d. *MediaWiki Laissez-Faire Control*

In the Laissez Faire DAC configuration, each group was given the sysop control rights: read, edit, limited delete, protect, and administrative permissions, with the exception of users' rights, big-delete, block, patrol, and protect. The popular service, WikiPedia, typically operates with Laissez-Faire controls. Those rights were tailored to enforce the rule of less privilege so that overall control remains with an administrator. The test plan is shown in Table 13 and Table 14 shows the results from testing.



Test	Procedure
Read	
	can all groups view pages
Editing	
	edit- allow all groups to edit unprotected pages
	createpage- allow all groups to create new pages
	createtalk- allow all groups to create of new talk pages
	move- allow all groups to rename the titles of unprotected pages
	createaccount- allow all groups to create new user accounts
	upload- allow all groups to create new images and files
	reupload- allow all groups to overwrite existing images and files
	reupload-shared- allow all groups to replace images and files from a shared repository
	upload_by_url- allow all groups to upload by entering the URL of an external image
Delete	
	delete- allow all groups to delete or undelete of pages
	bigdelete- allow all groups to delete pages with larger than $wgDeleteRevisionsLimit revisions
	deletedhistory- allow all groups to view deleted revisions, but not restoring
	mergehistory- allow all groups access to Special:MergeHistory, to merge non-overlapping pages.
Protect	
	allows all groups to lock a page to prevent edits and moves, and editing or moving locked pages
	block- prevents other groups from editing and registering new accounts, and autoblocking other users on the same IP address.
	blockemail- allow all groups to prevent other groups from using the Special:Emailuser interface when blocking.
	hideuser- allow all groups to hide the user/IP from the block log, active block list, and user list when blocking.
Admin	
	userrights- allow all groups to assignment or removal of all* groups to any user
	interwiki- allow all groups to change user rights on other wikis
	rollback- allow all groups one-click reversion of edits
	markbotedits- allow all groups rollback to be marked as bot edits
	patrol- allow all groups to mark edits as legitimate
	editinterface- allow all groups to edit the MediaWiki namespace, which contains interface messages
	editusercssjs- allow all groups to edit user's own monobook.css, monobook.js, ... Subpages
	hiderevision- allow all groups to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	deleterevision- allow all groups to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.
	siteadmin- allow all groups to lock and unlock the database (which blocks all interactions with the web site except viewing). Deprecated by default.
	import- allow all groups to import one page per time from another wiki.
	importupload- allow all groups user to import several pages per time from XML files.
	trackback- allow all groups removal of trackbacks
	unwatchedpages- allow all groups access to Special:Unwatchedpages, which lists pages that no user has watchlisted

Table 13. MediaWiki Laissez-Faire Test Plan (From: [26])



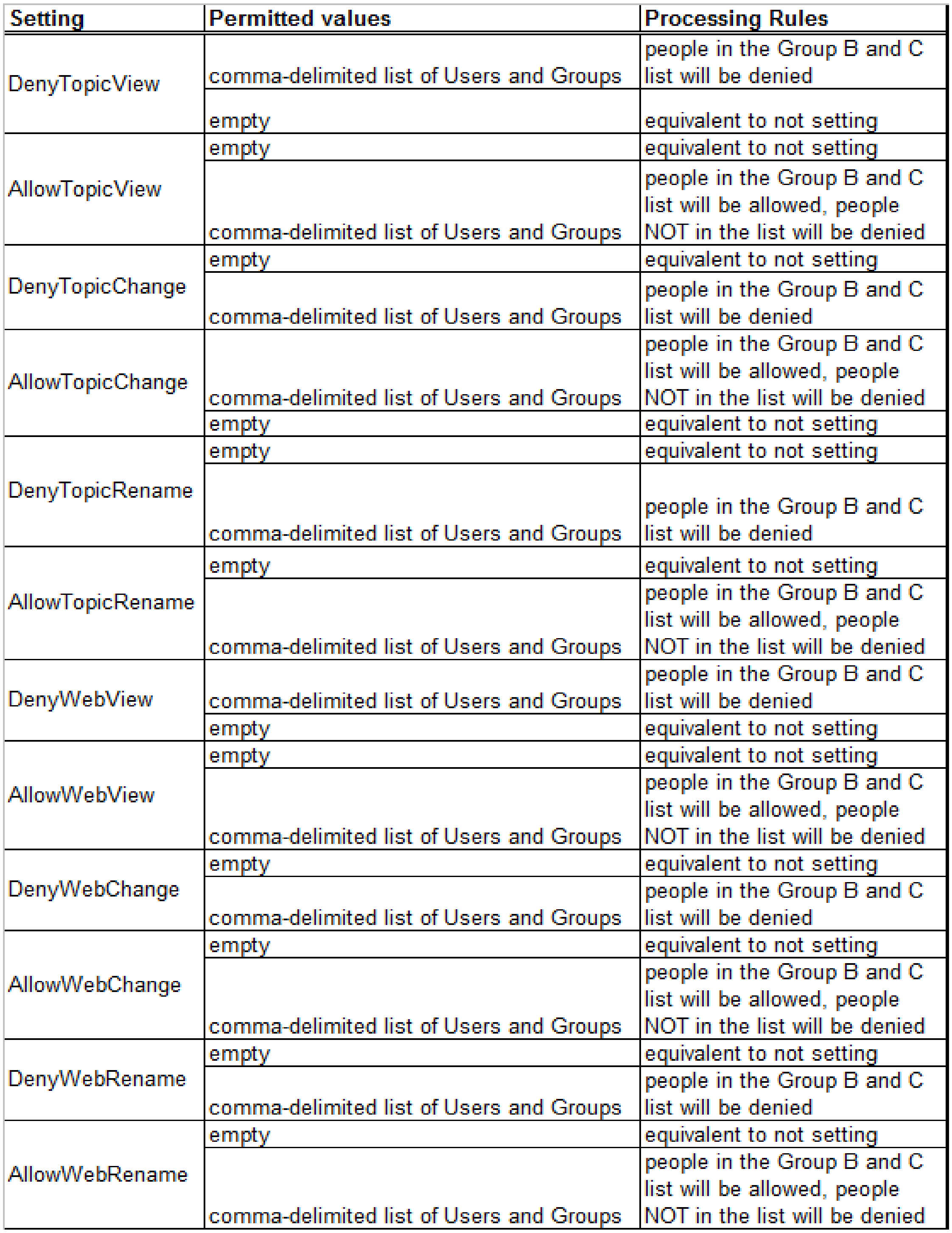
Test	Procedure	Result
Read		
	can all groups view pages	access
Editing		
	edit- allow all groups to edit unprotected pages	access
	createpage- allow all groups to create new pages	access
	createtalk- allow all groups to create of new talk pages	access
	move- allow all groups to rename the titles of unprotected pages	access
	createaccount- allow all groups to create new user accounts	access
	upload- allow all groups to create new images and files	access
	reupload- allow all groups to overwrite existing images and files	access
	reupload-shared- allow all groups to replace images and files from a shared repository	access
	upload_by_url- allow all groups to upload by entering the URL of an external image	access
Delete		
	delete- allow all groups to delete or undelete of pages	access
	bigdelete- allow all groups to delete pages with larger than $wgDeleteRevisionsLimit revisions	denied
	deletedhistory- allow all groups to view deleted revisions, but not restoring	access
	mergehistory- allow all groups access to Special:MergeHistory, to merge non-overlapping pages.	n/a
Protect		
	allows all groups to lock a page to prevent edits and moves, and editing or moving locked pages	access
	block- prevents other groups from editing and registering new accounts, and autoblocking other users on the same IP address.	denied
	blockemail- allow all groups to prevent other groups from using the Special:Emailuser interface when blocking.	n/a
	hideuser- allow all groups to hide the user/IP from the block log, active block list, and user list when blocking.	n/a
Admin		
	userrights- allow all groups to assignment or removal of all* groups to any user	denied
	interwiki- allow all groups to change user rights on other wikis	n/a
	rollback- allow all groups one-click reversion of edits	access
	markbotedits- allow all groups rollback to be marked as bot edits	access
	patrol- allow all groups to mark edits as legitimate	denied
	editinterface- allow all groups to edit the MediaWiki namespace, which contains interface messages	access
	editusercssjs- allow all groups to edit user's own monobook.css, monobook.js, ... Subpages	access
	hiderevision- allow all groups to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	deleterevision- allow all groups to prevent deleted revision information from being viewed by sysops and prevents sysops from undeleting the hidden info.	n/a
	siteadmin- allow all groups to lock and unlock the database (which blocks all interactions with the web site except viewing). Deprecated by default.	denied
	import- allow all groups to import one page per time from another wiki.	access
	importupload- allow all groups user to import several pages per time from XML files.	access
	trackback- allow all groups removal of trackbacks	access
	unwatchedpages- allow all groups access to Special:Unwatchedpages, which lists pages that no user has watchlisted	access

Table 14. MediaWiki Laissez-Faire Test Results (From: [26])

4. TWiki Discretionary Access Control Testing

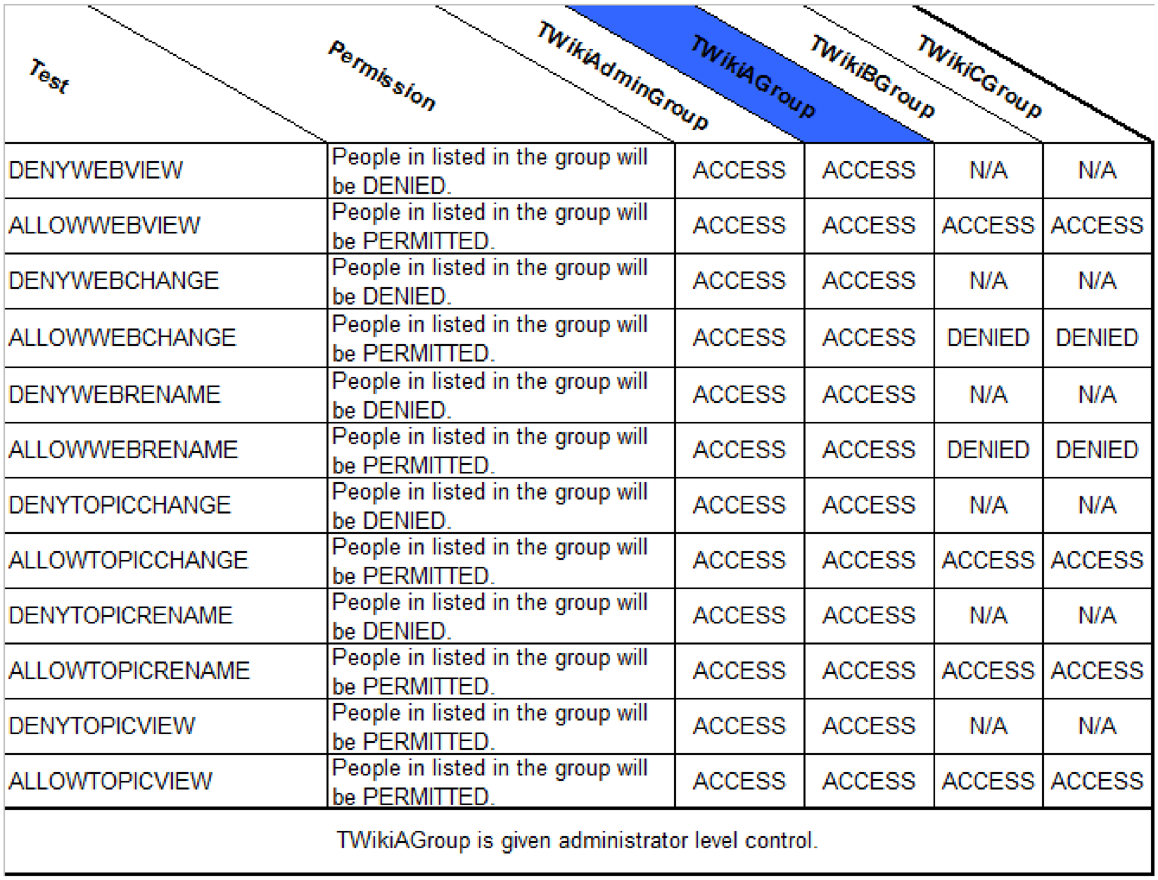
a. TWiki Hierarchical Control

Testing of hierarchical DAC configuration, users in TWikiAGroup were given administrative control rights, AllowTopicView, AllowTopicChange, AllowTopicRename, AllowWebView, AllowWebChange, and AllowWebRename. This test showed that TWikiAGroup could access topic and webs created by other groups, and TWikiAGroup could pass or upgrade permissions to allow TWikiBGroup and TWikiCGroup to perform one or all of the above permission modes. TWikiBGroup was given the right to AllowTopicChange and AllowTopicRename, AllowWebChange, AllowWebRename, which in turn gave TWikiBGroup the control to delegate permissions TWikiCGroup. Results are expressed either by ACCESS or DENIED, and N/A shows settings not applicable for this configuration. Below, Table 15 shows the test plan and Figure 8 shows the results.



Setting	Permitted values	Processing Rules
DenyTopicView	comma-delimited list of Users and Groups	people in the Group B and C list will be denied
	empty	equivalent to not setting
AllowTopicView	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be allowed, people NOT in the list will be denied
DenyTopicChange	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be denied
AllowTopicChange	comma-delimited list of Users and Groups	people in the Group B and C list will be allowed, people NOT in the list will be denied
	empty	equivalent to not setting
DenyTopicRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be denied
AllowTopicRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be allowed, people NOT in the list will be denied
DenyWebView	comma-delimited list of Users and Groups	people in the Group B and C list will be denied
	empty	equivalent to not setting
AllowWebView	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be allowed, people NOT in the list will be denied
DenyWebChange	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be denied
AllowWebChange	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be allowed, people NOT in the list will be denied
DenyWebRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be denied
AllowWebRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the Group B and C list will be allowed, people NOT in the list will be denied

Table 15. TWiki Hierarchical Test Plan (From: [23])



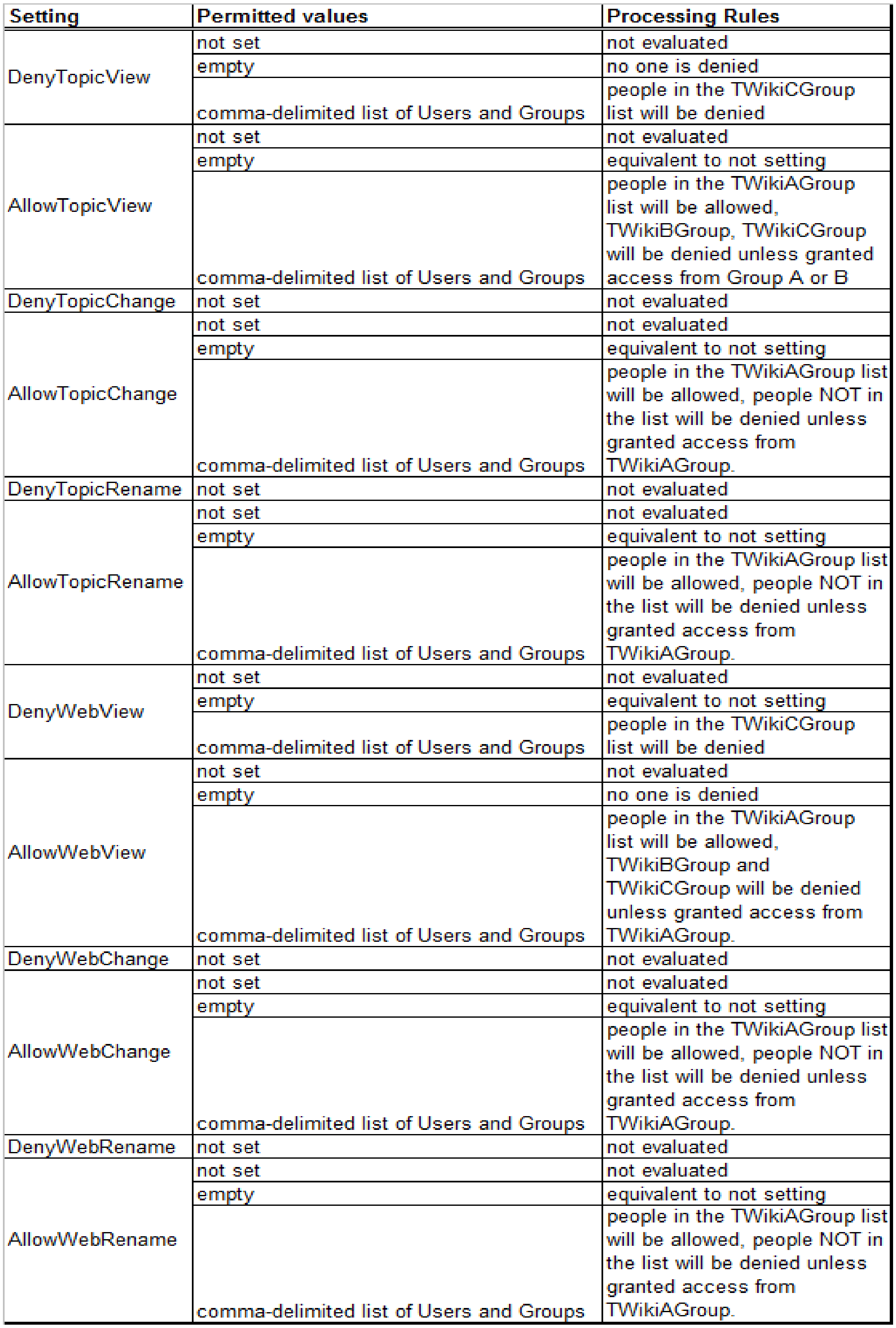
Test	Permission	TWikiAdminGroup	TWikiAGroup	TWikiBGroup	TWikiCGroup
DENYWEBVIEW	People in listed in the group will be DENIED.	ACCESS	ACCESS	N/A	N/A
ALLOWWEBVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYWEBCHANGE	People in listed in the group will be DENIED.	ACCESS	ACCESS	N/A	N/A
ALLOWWEBCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	DENIED	DENIED
DENYWEBRENAME	People in listed in the group will be DENIED.	ACCESS	ACCESS	N/A	N/A
ALLOWWEBRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	DENIED	DENIED
DENYTOPICCHANGE	People in listed in the group will be DENIED.	ACCESS	ACCESS	N/A	N/A
ALLOWTOPICCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYTOPICRENAME	People in listed in the group will be DENIED.	ACCESS	ACCESS	N/A	N/A
ALLOWTOPICRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYTOPICVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	N/A	N/A
ALLOWTOPICVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS

TWikiAGroup is given administrator level control.

Figure 10. TWiki Hierarchical Test Results (From: [23])

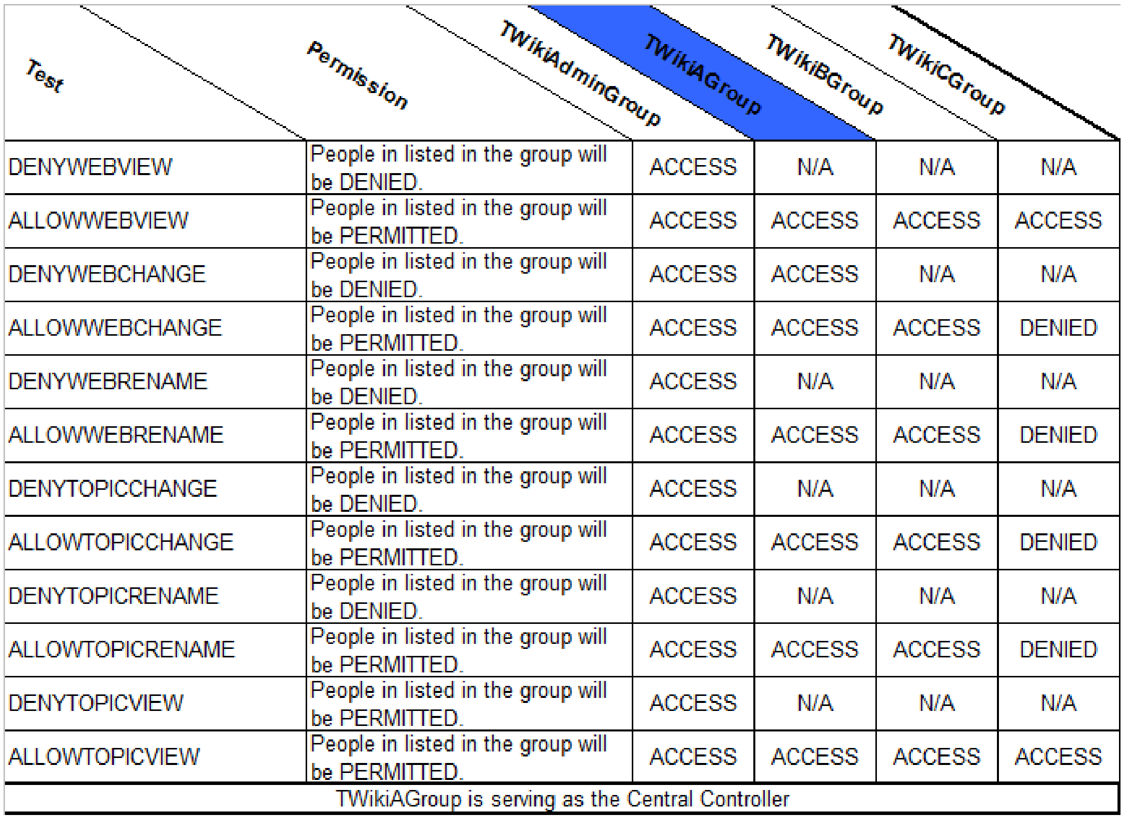
b. *TWiki Centralized Control*

TWiki centralized configuration shows that TWikiAGroup is the centralized controller. TWikiBGroup was granted permissions to view, change and rename mode permissions to webs and topics. Then TWikiBGroup could set permissions granting or denying control modes to TwikiCGroup. TWikiCGroup was denied access to change and rename, but was given the right to view. The setting of these mode permission were extremely challenging due to the denying control mode. For example, setting of the denying modes depending on your configuration could, in turn, deny TWikiBGroup access to the web. In order to overcome this, ensure controls are not set to the main web. Table 16 and Figure 9 show the test plan used and results.



Setting	Permitted values	Processing Rules
DenyTopicView	not set	not evaluated
	empty	no one is denied
	comma-delimited list of Users and Groups	people in the TWikiCGroup list will be denied
AllowTopicView	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the TWikiAGroup list will be allowed, TWikiBGroup, TWikiCGroup will be denied unless granted access from Group A or B
DenyTopicChange	not set	not evaluated
AllowTopicChange	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the TWikiAGroup list will be allowed, people NOT in the list will be denied unless granted access from TWikiAGroup.
DenyTopicRename	not set	not evaluated
AllowTopicRename	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the TWikiAGroup list will be allowed, people NOT in the list will be denied unless granted access from TWikiAGroup.
DenyWebView	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the TWikiCGroup list will be denied
AllowWebView	not set	not evaluated
	empty	no one is denied
	comma-delimited list of Users and Groups	people in the TWikiAGroup list will be allowed, TWikiBGroup and TWikiCGroup will be denied unless granted access from TWikiAGroup.
DenyWebChange	not set	not evaluated
AllowWebChange	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the TWikiAGroup list will be allowed, people NOT in the list will be denied unless granted access from TWikiAGroup.
DenyWebRename	not set	not evaluated
AllowWebRename	not set	not evaluated
	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people in the TWikiAGroup list will be allowed, people NOT in the list will be denied unless granted access from TWikiAGroup.

Table 16. TWiki Centralized Test Plan (From: [23])

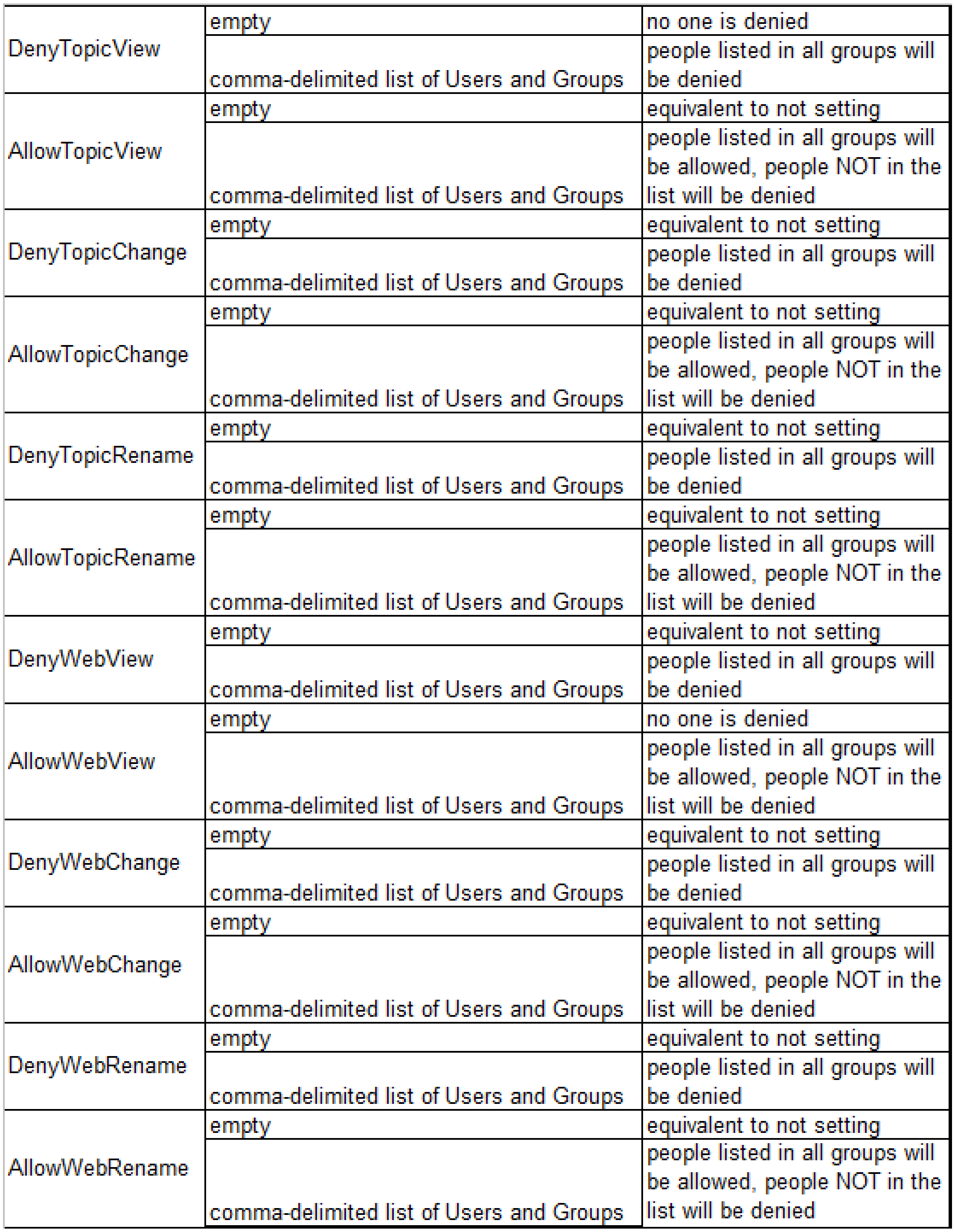


Test	Permission	TWikiAdminGroup	TWikiAGroup	TWikiBGroup	TWikiCGroup
DENYWEBVIEW	People in listed in the group will be DENIED.	ACCESS	N/A	N/A	N/A
ALLOWWEBVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
DENYWEBCHANGE	People in listed in the group will be DENIED.	ACCESS	ACCESS	N/A	N/A
ALLOWWEBCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	DENIED
DENYWEBRENAME	People in listed in the group will be DENIED.	ACCESS	N/A	N/A	N/A
ALLOWWEBRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	DENIED
DENYTOPICCHANGE	People in listed in the group will be DENIED.	ACCESS	N/A	N/A	N/A
ALLOWTOPICCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	DENIED
DENYTOPICRENAME	People in listed in the group will be DENIED.	ACCESS	N/A	N/A	N/A
ALLOWTOPICRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	DENIED
DENYTOPICVIEW	People in listed in the group will be PERMITTED.	ACCESS	N/A	N/A	N/A
ALLOWTOPICVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
TWikiAGroup is serving as the Central Controller					

Figure 11. TWiki Centralized Test Results (From:[23])

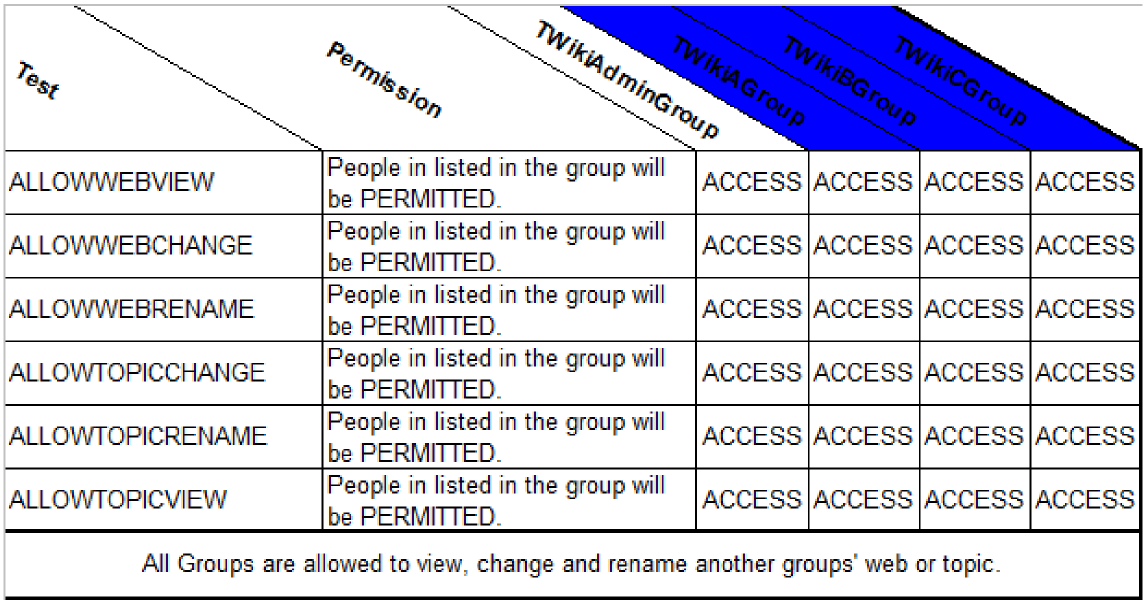
c. *TWiki Laissez-Faire Control*

The Laissez test plan and results show how all permissions modes was assigned to TWikiAGroup, TWikiBGroup and TWikiCGroup, with the exception of deny. Each group was given control rights to view, change, and rename the webs or topics. This configuration allows each group to have total access among all groups. Table 17 and Figure 10 show test plan and results.



DenyTopicView	empty	no one is denied
	comma-delimited list of Users and Groups	people listed in all groups will be denied
AllowTopicView	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be allowed, people NOT in the list will be denied
DenyTopicChange	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be denied
AllowTopicChange	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be allowed, people NOT in the list will be denied
DenyTopicRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be denied
AllowTopicRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be allowed, people NOT in the list will be denied
DenyWebView	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be denied
AllowWebView	empty	no one is denied
	comma-delimited list of Users and Groups	people listed in all groups will be allowed, people NOT in the list will be denied
DenyWebChange	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be denied
AllowWebChange	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be allowed, people NOT in the list will be denied
DenyWebRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be denied
AllowWebRename	empty	equivalent to not setting
	comma-delimited list of Users and Groups	people listed in all groups will be allowed, people NOT in the list will be denied

Table 17. TWiki Laissez-Faire Test Pan (From: [23])



Test	Permission	TWikiAdminGroup	TWikiAGroup	TWikiBGroup	TWikiCGroup
ALLOWWEBVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
ALLOWWEBCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
ALLOWWEBRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
ALLOWTOPICCHANGE	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
ALLOWTOPICRENAME	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS
ALLOWTOPICVIEW	People in listed in the group will be PERMITTED.	ACCESS	ACCESS	ACCESS	ACCESS

All Groups are allowed to view, change and rename another groups' web or topic.

Figure 12. TWiki Laissez-Faire Test Results (From: [23])

d. TWiki Ownership Control

The owner configuration could not be configured by using the web browser interface. Setting up configuration control modes requires the exploration of code associated within the data directory files. TWiki ownership testing is beyond the scope of the thesis, therefore, this configuration was not evaluated.

C. CONOPS IMPLEMENTATION

1. MediaWiki Implementation

Based upon the requirements, the fine granularity of discretionary access control, and testing results, we determined that MediaWiki would be best suited for scenario 2, JTF-PEAK. The JTF-PEAK scenario requires a traditional hierarchical structure. MediaWiki allows the flexibility in establishing groups. This can prove beneficial for each Component Commander when delegating access permissions to subordinates for plans and operations in order to preserve operation security (OPSEC) on the battlefield. Moreover, each Component Commander has the capability to establish sub-groups (i.e., Weapons, Comms) within their own organization to represent their existing organizational structure. Along with the capability to handle millions of images and pages, MediaWiki has the potential to enhance communications bandwidth. In addition, planning could be conducted across multiple geographical areas where operations can be conducted and orders can be executed in near-real-time. MediaWiki could aid in decreasing the time between issuing orders and the execution of those orders. In doing so, MediaWiki helps to flatten the C2 structure, thus aids in collaboration across all services and can assist in increasing combat effectiveness.

Additionally, based on the need to collaborate with coalition partners, the security features, are unquestionably robust enough to be implemented in a multinational CONOPS environment. For instance, information sharing can be completely open or restricted based upon the “need-to-know” access. The software security versatility would allow each group to have control of its associated groups pages or files and change their permissions based upon current operations.

MediaWiki possess the capability to meet JTF-TEAK CONOPS scenario with fine granularity in DAC security, its usability to non-programmers, and its ability to enhance file sharing and data management makes it idea for the JTF-TEAK scenario. Although MediaWiki DAC is suitable for JTF-TEAK environment, DAC alone cannot ensure perfect security of information.

2. TWiki Implementation

TWiki discretionary access control implementation is suitable for use in scenario 1, JIOC-X. TWiki display DAC flexibility needed to set up groups and segregate information based upon the *need to know.* Such flexibility is required for this particular CONOPS, based upon inter-agency and inter-services intelligence communities. This in turn would ease concerns about whether a particular CoP processes the clearance to evaluate, disseminate, or share intelligence information to organizations such as fire departments, local police departments or emergency assistance agencies in the event of a terrorism attack.

D. SUMMARY

This chapter described the methodology selection process, experimentation procedure, and gave implementation suggestions of possible use for MediaWiki and TWiki, respectively, in two scenarios relevant tot he DoD communities of practice. We next analyze these results with a comparative analysis of MediaWiki and TWiki discretionary controls in a CONOPS environment.

VI. COMPARATIVE ANALYSIS

The similarities and contrasts between MediaWiki's and TWiki's discretionary access policies have been tested and described in Chapter III. Although, the two wiki engines have the ability to configure hierarchical, centralized, ownership and laissez-faire control policies, their applicability for a secure CONOPS environment rarely differ. Their DAC policy configurations, modes of control, and DAC problems were discussed in this analysis.

A. ANALYSIS

1. Discretionary Access Control

In order to configure MediaWiki to meet DAC requirements for a secure CONOPS environment, two security configuration methods were tested. The first effort of establishing groups is not easily understood to a non-programmer as compared to a programmer. Groups had to be created by changing the Localsettings.php file, which can be done only by an administrator with super user control rights. Once the groups were created, they would have to be tailored based upon what DAC control configuration a particular CONOPS required. The second method in configuring DAC control permissions to groups was the use of the XAMMP program interface. Here, groups would be created, once created, rights could be assigned. Use of this method does not allow an administrator to customize group rights based upon a need for access. Furthermore, all groups would in turn have a standard level of access. This method is not recommended when attempting to establish control for use in a secure CONOPS environment.

As compared to MediaWiki, TWiki groups can be created either through root directory using the Linux interface or through the web server interface. Using the Linux interface requires super user control rights. Coding each configuration and creating groups could be difficult and require programmer level expertise. According to TWiki configuration tips, this is not the recommended method of configuring DAC for groups. However, improved Linux versions have a graphic user interface (GUI) capability to manage users and groups.

In contrast to using the Linux interface, the web server interface was more lucid when attempting to establish groups and when configuring DAC rights. For example, a member from the administrative group (without root level rights) could tailor control rights by clicking on the web or topic edit button. After editing, code(s) could be changed based upon the required DAC configuration for a particular CONOPS.

2. Modes of Control

Additionally, as compare to maintainability and usability TWiki ensures is user-friendly, and has a solid support structure. Due to the reliability and available to obtain timely and accurate information the need for a solid support structure is paramount. TWiki has a myriad of contributors who ensure the software remains up-to-dates, but also ensures critical security concerns are being addressed.

MediaWiki possess different mode control and access permissions rights as compared to TWiki. MediaWiki has four modes of control rights and forty access permissions. Those control right modes are user, bot, sysop, and bureaucrat and access permissions, most commonly known as read, edit, and delete, just to name a few. Assigning rights to meet a secure CONOPS environment would be simple for most administrators. For example, the Localsetting.php file, which was accessed by an administrator, could easily identify what rights needed to be assigned to a certain group. Due to over forty permission rights, MediaWiki also proved to be more flexible, although certain permission rights are defaulted based on which MediaWiki role is held. For instance, if a CONOPS called for sysop users to have “userrights” control, the administrator has the flexibility to assign that right to that user, a right usually held by bureaucrat mode of control.

In contrast to MediaWikis' many control modes, TWiki possess two control modes: administrator or registered user. In addition to control modes, TWiki also has three access permission rights. Those rights are a combination of twelve deny and allow access permissions to view, change or rename topics or webs. Having only three standard permission right policies, TWiki inherently makes it harder to configure for a secure CONOPS environment. For example, ownership control configuration could not be effectively implemented unless some advance programming methods are used. Moreover, although a user created the topic or web, the change and rename permission rights would still allow a non-owner a way to access and edit the topic created by the owner. In addition to having the rename right, this right essentially overrides the change right. Therefore, denying a user the change right would disallow that user's access to the main web or topic of interest.

MediaWiki and TWiki demonstrated similar advantages in terms of flattening the command and control structure. Both helps to elude organizational bureaucracy to foster expedite, accurate, and timely decision-making on the battlefield. Its social aspects allows decision-makers at all levels of the organizational structure to more effectively employ creative human capital to increase operational and combat effectiveness across international and non-governmental domains [4].

Another similarity between MediaWiki and TWiki an administrator's capability to tailor discretionary access control for a particular CONOPS scenario. Flexibility during implementation implicitly adds real-value when configuring DAC for various relational database. For example, the sharing of intelligence with two opposing coalition partners could prove to be challenging. However, the flexibility to tailor permissions to relational databases promotes good diplomacy in the sharing of intelligence.

3. Discretionary Access Control Problems

Fundamentally, the discretionary access controls as implemented in both MediaWiki and TWiki do not discriminate between those who administer the system and those users who have need to access sensitive content. Particularly in a coalition or interagency collaboration environment, it is likely that the system administrators will need access at the file-level with no need (or clearance) to view file contents. This is a fundamental flaw in the Wiki perspective on DAC.

Additional, serious, implementation issues abound in these systems. MediaWiki permits information to be leaked by virtue of the simple page-specific extensions to restrict user access. For example, MediaWiki caches one version of a page and then serves that page to everyone without rechecking to see if the next user has the proper rights. This could, in turn, allow data from users with higher rights to be viewed by a user with fewer rights; this potential security compromise makes MediaWiki unsuitable for secure collaboration. Turning off the cache would correct the problem, but will deny authorized users access as well. TWiki shows a similar read-up cache problem, with similar security implications. *One major finding in this work is that these Wiki implementations could easily allow a security compromise in its present implementation*.

Additional significant configuration issues were demonstrated in experimental runs. With MediaWiki in particular, the default configuration places database passwords in a plain text file located on the same server as the MediaWiki installation. Any compromise of the system serving Wiki pages means an almost-certain compromise of the entire installation.

In their current implementations, both TWiki and MediaWiki offer attractive features for distributed collaboration. Both, however, are completely unsuitable for secure work.

B. SUMMARY

The use of wiki engines, MediaWiki and TWiki put forward solutions in breaking down culture barriers between different communities of practice. This thesis highlights wiki discretionary access control capabilities and underlining problems for use in a secure DoD CONOPS environment. Although discretionary access control was the focus, discretionary access control only is not the solution to securing information in a CONOPS environment.

In this research, we have explored the use of new collaborative technology, the wiki, to accomplish a primary DoD task, drafting a concept of operations (CONOPS) document. In particular, we have examined the access control aspects of this technology with respect to the practice of CONOPS development. Our results suggest that the wiki may indeed be a force multiplier, and represent a significant advance not only in technology but also in organizational thinking. Wiki collaboration helps to put unprecedented power in the hands of decision-makers on all level of the chain of command. This thesis gave detailed comparative analysis of each DAC policy configuration and modes of control was given to aid decision-makers in choosing a wiki engine for a secured CONOPS environment. Wiki engines technology is not the sole solution to eliminating communication and cultural stovepipes when collaborating information, but rather a piece of the puzzle to a complex problem.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION AND FUTURE WORK

In this work, we have examined the discretionary access control mechanisms of two popular wiki engines with respect to common DoD collaborative tasks. Specifically, we have examined TWiki and MediaWiki for use in producing a secure CONOPS in operationally relevant scenarios. Results from this work suggest that both engines in particular, and the wiki paradigm in general, offer significant opportunities for future DoD collaborative work, particularly with coalition and non-governmental partners. Experimental results demonstrate that Media and TWiki (with the exception of centralized control in TWiki) possess a sufficiently fine granularity of discretionary access control to support these collaborative tasks.

Future experiments could explore similar collaborative engines built with slightly different paradigms. For example, another fielded technology is the Zimbra Collaboration Suite (ZCS), a suite that supports email and group calendars using an Ajax web interface [33].

Although the CONOPS represents a common task for DoD agencies, other significant, collaborative tasks could be explored with the wiki. For example, many points in the Air Operations Center (AOC) Air Tasking Order (ATO) process might benefit from a wiki tool. Vehicle maintenance and supply communities, too, may benefit from a distributed, wiki tool for their day-to-day operation.

With respect to discretionary access controls within wiki technology, future research could address establishing page specific extension coding schemes to implement a robust DAC. Currently the page extension allows page read-up access to less privileged users prior to rechecking the user rights. Another avenue of research would be the separation of administrative (system-specific) rights from the rights to view and change content (information-specific rights). Particularly within DoD environments, in many cases the system administrators have no compelling need to view the content owned and maintained by their supported users. One rich area for research would be to determine whether compartments can be established for web administrators to preserve the confidentiality and authenticity of sensitive information.

Experimental results suggest the wiki framework may be an appropriate tool for many DoD collaborative tasks. Looking at specific wiki implementations, however, suggest that there is still much work to be done with discretionary access control to meet DoD environment security requirements.

APPENDIX A. MEDIAWIKI INSTALLATION AND CONFIGURATION

There are several versions of MediaWiki to choose from, therefore, choosing a version may depend upon your CoP organizational structure or desired capabilities. For the purpose of this testing, MediaWiki version 1.12.0 and Windows XP operating system is used. It is important that you check to be certain your system meets the minimum requirements (apache http web server v2.2, MySQL4 and PHP5) prior to installing. The following are steps to guide you through installation and configuration of MediaWiki.

A. INSTALLING MEDIAWIKI VERSION 1.12.0[1] UNDER WINDOWS

1. Creating a Testing Environment

Step 1. Download the latest version of *XAMPP for Windows* from site URL http://www.apachefriends.org/en/xampp.html

Step 2. Execute the *.exe file* and complete download.

Step 3. Select a language the most appropriate to you and click Next.

Step 4. Read and Agree to the license information.

Step 5. Click the Install to start the process. Note: the program creates the folder *C:/apachefriends/xampp*.

Step 6. Run the file *xampp_start.exe*

Step 7. Open a browser and enter URL http://127.0.0.1 or localhost. Note: The *XAMPP* environment start page should appear. At this point *MySQL* or *PHP* have been activated.

[1] Apache Friends, http://www.apachefriends.org/en/xampp.html.

2. Testing the environment

Step 8. Create subdirectory *xampp/htdocs/test*

Step 9. Open the test editor and save the following code in the *index.php* in the test folder.

```
<html>
        <head>
                <title>This is a test</title>
        </head>
        <body>
                <p>Your own pages are displayed here
                <p><?php echo “PHP running” ?>
</html>
```

Step 10. If using Windows XP operating system. Go to *Control Panel*, then *Tools*, then *Folder Options*, then *View*. Deactivate(uncheck) the option “*Hide extension for known file types”.*

3. Installing MediaWiki to a Local System

Step 11. Download Media latest version from site *www.mediawiki.org*.

Step 12. Copy compressed file to directory *xampp/htdocs* and unpack with *FilZip* under Windows. Note: Rename the directory *mediawiki*

Step 13. In your browser, type *http://localhost/mediawiki.*

Note: You should see the message, *“You’ll have to set the wiki up first.”* However, installation is not complete.

Step 14. Set the username and password of the administrator account. Note: Change the name to anything other than “*WikiSysop.*”

Step 15 Set the root password in *MYSQL*

Step 16 Press the INSTALL button

Step 17 Copy the file *LocalSettings.php* located in directory *xampp/htdocs/mediawiki/config* to directory *xampp/htdocs/mediawiki*, located one level higher

Step 18 Last; open the Wiki URL in the browser and the you should the main page of *Mediawiki.*

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. TWIKI INSTALLATION AND CONFIGURATION

A. INSTALLING TWIKI VERSION 4.2.0 UNDER WINDOWS[1]

1. Creating a Testing Environment

Step 1. Download the latest version of *XAMPP for Windows* From site URL http://www.apachefriends.org/en/xampp.html

Step 2. Execute the *.exe file* and complete download.

Step 3. Select a your language and click Next.

Step 4. Read and Agree to the license information.

Step 5. Click the Install button to start the process. Note: the program creates the folder *C:/apachefriends/xampp*.

Step 6. Run the file *xampp_start.exe*

Step 7. Open a browser and enter URL *http://local host*. Note: The *XAMPP* environment start page should appear. At this point *MySQL* or *PHP* have been activated.

2. Testing the Environment

Step 8. Create subdirectory *xampp/htdocs/test*

Step 9. Open the test editor and save the following code in the *index.php* in the test folder.

```
<html>
<head>
        <title>This is a test</title>
</head>
```

```
<body>
        <p>Your own pages are displayed here
        <p><?php echo “PHP running” ?>
        </html>
```

Step 10. Go to *Control Panel*, then *Tools*, then *Folder Options*, then *View*.

Deactivate(uncheck) the option “*Hide extension for known file types”.*

3. Installing TWiki Version 4.2.0 under Windows

Step 11. Set up download Cygwin from site http://www.cygwin.com and select “Install from the Internet.” Note: Make sure the Default Text File Type” is set to Unix and Cygwin should be set up for all users.

Step 12. Enter the directory where files will be stored, supply internet connection information, and select server a server from the list.

Step 13. Ensure each source file shown below is selected and click *Continue*.

Archive unzip – Unpack .zip files

Base bash – command line interpreter under Unix.

coreutils – Here: tools for editing text files.

diffutils - Finds differences between files.

grep – Searches for specific patterns in character strings and files.

gzip – GNU compression utility.

Tar – GNU archiving untility

Develbinutils – GNU assembler and linker.

gcc – C compliler

make – Make Tool

pcre – perl library of regular expressions.

rcs – versioning software

Editors nano – Simple text editor.

Interpreters perl – Interpreter for the Perl script language.

Libs w32 – Access to Windows functions.

Net ncftp – FTP program

Web wget – for downloading files from internet

Step 14. Last, place Icon on desktop and Cygwin install is complete.

4. Configuring Perl

Step 15. Start up Cygwin and enter export TEMP=/c/temp

Step 16. Type *cpan* to open CPAN program and answer all questions prior to starting the configuration.

Step 17 Type the *install* commands below and download 3 TWiki required modules. After, installations Exit CPAN with the command *exit.*

install Net::SMTP

install Digest::SHA1

install MIME::Base64

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1]. Ebersbach Anja, Glaser Markus and Heigl Richard, “Wiki Web Collaboration” Springer-Verlag Berlin Heidelberg, New York 2006.

[2]. Carole S. Jordan, Dr. Deborah Downs, Grant Wagner, Steve LaFountain, and Dr. Dixie B. Baker, “A Guide to Understanding: Discretionary Access Control in Trusted Systems.” National Computer Security Center. Fort George Meade, Maryland. September 1987.

[3]. E. Wenger, R. McDermott, and W.M. Snyder, Cultivating communities of practice: A guide to managing knowledge. Boston, MA: Harvard Business School Press, p. 4. 2002.

[4]. Don Tapscott and Anthony D. Williams, “Wikinomics: How Mass Collaboration Changes Everything”. Penguin Group Inc., New York, New York 2006.

[5]. J. Arguello, et al., Talk to me: Foundations of successful individual-group interactions in online communities. In Proc. CHI. 2006.

[6]. W. Emigh, and S.C. Herring, “Collaborative Authoring on the Web: A genre analysis of online encyclopedias.” In Proc. HICSS. 2005.

[7]. J.J. Cadiz, Gupta Anoop and Jonathan Grudin, “Using Web Annotations for Asynchronous Collaboration around Documents.” Proceeding of ACM Conference on Computer-Supported Cooperative Work, 2000.

[8]. Paul Dourish, and Victoria Bellotti, “Awareness and Coordination in Shared Workspaces.” Proceedings of the ACM Conference of Computer-Supported Cooperative Work. (1992)

[9]. Judith Donath, and Niel Robertson, “The Sociable Web” Proceedings of the Second International WWW Conference. Chicago, IL. October, 1994.

[10]. David A. Smith, Alan Kay, Andreas Raab, and David P. Reed, “Croquet Collaboration System Architecture.” In IEEE First Conference on Creating, Connecting and Collaborating through Computing, 2003.

[11]. David Ferraiolo and Rick Kuhn, “Role-based access controls.” In 15th NIST-NCSC National Computer Security Conference, pp. 554-563, 1992.

[12]. Arun Kumar, Neeran Karnik, and Girish Chafle, “Context Sensitivity in Role-based Access Control.” SIGOPS Oper. Syst. Rev., 36(3):53-66, 2002.

[13]. Teresa F. Lunt, “Access Control Policies: Some Unanswered Questions.” Computer Science Laboratory, SRI International. Menlo Park, CA. 1989.

[14]. David S. Alberts and Richard E. Hayes, “The Future of Command and Control: Planning Complex Endeavors.” Library of Congress, April 2007.

[15]. Mark Minasi, Christa Beveridge, C.A. Callahan and Lisa Justice. “Mastering Windows Server 2003, Fourth Edition.” SYBEX Inc.. San Francisco, CA 2003.

[16]. United States Joint Forces Command, “Joint Intelligence Operations Capability – Transformational (JIOC-X) Enterprise Concept of Operations Paper Version 0.4.” August 2006.

[17]. Graphics Naval War College, 2001 NSDM Situation Template, 14 April 2006.

[18]. Michael M. Sweeney, Naval Postgraduate School M.S. thesis. June 2002.

[19]. Wikimatrix, “http://www.wikimatrix.org/,” last viewed 9 January 2008.

[20]. MediaWiki, “http://www.mediawiki.org/wiki/Category: Configure,” last viewed 10 April 2008.

[21]. TWiki, “http://www.twiki.org/,” last viewed 05 May 2008.

[22]. Randy Hess, Randy retired US Navy. Personal correspondence. 29 Nov 2007.

[23]. TWiki, “http://twiki.org/cgi-bin/view/TWiki/TWikiAccessControl.” last viewed 16 May 2008.

[24]. Wiki, “http://c2.com/cgi/wiki? TopTenWikiEngines.” last viewed 10 January 2008.

[25]. Galois Inc., http://www.galois.com/xdomain.php, last viewed 23 May 2008.

[26]. MediaWiki, http://www.mediawiki.org/wiki/Manual:User_rights, last viewed 16 May 2008.

[27]. David S. Alberts, John J. Garstka and Frederick P. Stein, “Network Centric Warfare: Developing and Leveraging Information Superiority, 2nd Edition (Revised). Washington, D.C. 1999.

[28]. MediaWiki, “http://www.mediawiki.org/wiki/MediaWiki.” last viewed 3 June 2008.

[29]. Wikipedia, “http://en.wikipedia.org/wiki/Main_Page.” last viewed 3 June 2008.

[30]. Guidance from National Computer Security Center, “A Guide to Understanding Discretionary Access Control in Trusted Systems," http://www.fas.org/irp/nsa/rainbow/tg003.htm. last viewed 3 June 2008.

[31]. Jennifer Vesperman, “Version Control and Source Code Management: Essential CVS” First Edition, June 2003.

[32]. David S. Alberts and Richard E. Hayes, “Understanding Command and Control,” Washington, DC. 2006.

[33]. Wikipedia, http://en.wikipedia.org/wiki/Zimbra.” last viewed 11 June 2008.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California